Farnaz.mohammadjeh@gmail.com

سیستم‌های عددی:

$(A)_r = (a_n r^n , \dots , a_2 r^2 , a_1 r^1 , a_0 r^0 , a_{-1} r^{-1} \; a_{-2} r^{-2} \; \dots \; a_{-m} r^{-m})$

تبدیل از مبنای ۱۰ به r ← تقسیمات متوالی در مبنای r

$(786)_{10} = (1100010010)_2$

$(4768)_{10} = (12\,9\,0)_{16}$

۷۸۶ ÷ ۲ → ۳۹۳ ÷ ۲ → ۱۹۶ ÷ ۲ → ۹۸ ÷ ۲ → ۴۹ ÷ ۲ → ۲۴ ÷ ۲ → ۱۲ ÷ ۲ → ۶ ÷ ۲ → ۳ ÷ ۲ → ۱

باقیمانده‌ها: ۰، ۱، ۰، ۰، ۱، ۰، ۰، ۰، ۱، ۱

۴۷۶۸ ÷ ۱۶ = ۲۹۸ ، باقیمانده ۰

۲۹۸ ÷ ۱۶ = ۱۸ ، باقیمانده ۱۰

۱۸ ÷ ۱۶ = ۱ ، باقیمانده ۲

مبناهایی که ارقامش ۲ رقمی شده

10 → a 11 → B 12 → C 13 → D

14 → E 15 → F

در مبنای ۲ فقط به جای تقسیمات متوالی می‌بینیم که از

32 16 8 4 2 1

چه مقداری برداریم (بزرگ‌ترین مقدار) از طریق ارزش‌گذاری حل می‌کنیم.

$(15.73)_{10} = ($

یعنی می‌بینیم چی داریم که ۱ بزاریم و چی نداریم که ۰ بزاریم.

→ $(1111.101\dots)_2$

$.73 \times 2$

$1.46 \rightarrow 0.46 \times 2 = 0.92 \rightarrow 0.92 \times 2 = 1.84$

PAPCO

Scanned by CamScanner

Subject:
Date

تبدیل ۲ به ۱۰ ←

$(\underset{16\ 8\ 4\ 2\ 1}{11010}.\overset{2^{-1}}{1})_2 = (۲۶.۵)_{10}$

$(715)_8 = (\overset{۷\times۶۴+۸+۵}{461})_{10}$

تبدیل از r_1 به r_2

$(A)_{r_1} = (\quad)_{r_2}$

$\searrow (\quad)_{10} \nearrow$

اگر r_1، r_2 نسبت توانی داشتند می شه مستقیم تبدیل کرد.

برای مثال ← از مبنای ۸ به ۲ ←

$(621)_8 = (110010001)_2$

یعنی ۸ توان ۳ از ۲ است یعنی هر خانه مبنای ۸ معادل ۳ خانه مبنای ۲

۱۶ ۸ ۴ ۲ ۱

به طور مشابه ۱۶ به ۲.

$(A0F5.2)_{16} = (1010\ 0000\ 1111\ 0101\ .\ 0010)_2$

مبنای ۲ به ۸ ← از همین گروه های سه تایی ایجاد می کنیم. (از سمت راست)

$001\,101\,110 . 101\,100 = (۱۵۶.۵۴)_8$

$001\,110\,111 . 110_2 = (۱۶۷.۶)_8$

[illegible] ۲ به ۱۶ ←

$(1111\,0001\,0001 . 1010\,1000) = (F۱۱.A۸)_{16}$

Subject:
Date

جمع

$(A019)_{16}$ + $(58A)_{16}$ = $(A5A3)_{16}$	11011011 + 1011111 = 100111010	اگر جمع ارقام از عدد مبنا بیشتر شد مبنا را به عنوان عدد نقلی به رقم بعدی می دهیم اضافه کم می کنیم

تفریق

$(A0E5)_{16}$ − $(BCF)_{16}$ = $(9446)_{16}$	$(10110101)_2$ − 1011100 = 1011001

مکمل: $r-1$ ، r

$A=(a_n \dots a_2 a_1 a_0)$

$r-1$ مکمل → $(r-1-a_n \dots r-1-a_2 \quad r-1-a_1 \quad r-1-a_0)$

یعنی کل ارقام را از $r-1$ کم می کنیم — در مبنای ۲ ، ۰ها ۱ و ۱ها ۰

مکمل ۲ ← اولین رقم غیر صفر از r بقیه از $r-1$ از سمت راست. به جز صفرها

$A=(a_n \dots a_1 a_0)$

↳ $(r-1-a_n \dots r-1-a_n \quad r-a_n)$

★ مکمل ۲ یک واحد از مکمل $r-1$ بزرگتر است.

استفاده از تفریق برای مکمل پیره : ($r-1$)

۱) تعداد ارقام را یکسان می کنیم

۲) $r-1$ (عدد دوم) مکمل

۳) $A+(r-1$ مکمل $B)$ اصلاح + است

$A=20 \quad B=5 \quad \rightarrow \quad A-B=15$

```
   10100
 + 11010
 -------
[1] 01110   (5)
 +      1   اصلاح
 -------
    1111
    +15
```

علامت نقلی ← [1] ؛ حاصل تفریق

PAPCO

Subject:
Date

وقتی رقم نقلی آخر ۰ شد یعنی جواب منفی است.

۴) رقم نقلی → ۱ + → جواب آخر → S+1

→ ۰ − → " → مکمل r-1

مکمل r ←

A=18 B=4 → A−B = 14

۱) تعداد ارقام را یکسان می کنیم

۲) مکمل r

۳) A + (مکمل r B)

$$\begin{array}{r} 10010 \\ +\ 11100 \\ \hline \boxed{1} \leftarrow 01110 \\ S \end{array}$$

+ یعنی حاصل تفریق

۴) اصلاح → آخر رقم نقلی مرتبه آخر ۱ بود → جواب خود S

→ " ۰ " " → یعنی حاصل تفریق − است → S+ (مکمل r)

جلسه ۲

کد کردن اطلاعات: BCD → Binary Code Decimal

هر رقم ← ۴ بیت

8 5 3

1	0	0	0	0	1	0	1	0	0	1	1
8	4	2	1	8	4	2	1	8	4	2	1

BCD (6421)

1	0	1	0	0	1	0	1	0	0	1	1
6	4	2	1	6	4	2	1	6	4	2	1

PAPCO

Scanned by CamScanner

Subject:
Date

EXCESS-3 ← یعنی اول تمام ارقام را هر کدام با ۳ جمع می‌کنیم و بعد تبدیل به BCD می‌کنیم

	8	5	3		7	8	4
+3	11	8	6		10	11	7
	1011	1000	0110		1010	1101	0111

Gray ← باید فقط در یک بیت با هم متفاوت باشند.

تبدیل کد باینری به گری ← چپ‌ترین مقدار همان
ارقام را دوتا دوتا XOR

$(1101001)_B = (?)_G$

$(1011101)_G$

$(10111000)_B = (?)_G$

$(11100100)_G$

00	0
01	1
10	1
11	0

جبر بول

$a+b=b+a$ | $a \cdot b = b \cdot a$

$a+a=a$ | $a \cdot a = a$

$a+\bar{a}=1$ | $a \cdot \bar{a} = 0$

$a+1=1$ | $a \cdot 0 = 0$

$a+0=a$ | $a \cdot 1 = a$

★ $a+(b \cdot c)=(a+b) \cdot (a+c)$ | ★ $a \cdot (b+c) = (a \cdot b)+(a \cdot c)$

$\overline{a+b} = \bar{a} \cdot \bar{b}$ | $\overline{a \cdot b} = \bar{a}+\bar{b}$

Scanned by CamScanner

Subject:
Date

$a + ab = a(\underset{}{1+b}) = a \cdot 1 = a$

$a \cdot (a+b) = a + ab = a$

$a + \bar{a}b = (a + \bar{a}) \cdot (a+b) = a + b$?

$a \cdot (\bar{a} + b) = (a \cdot \bar{a}) + (ab) = ab$

$\bar{A}B(\bar{D} + \bar{C}D) + B(A + \bar{A}CD) = B(\bar{A}\bar{D} + \bar{A}\bar{C} + A + CD) = B \cdot 1 = 1$

$(D + \bar{D})(\bar{C} + \bar{D})$ $\quad A + CD \quad$ $A + \bar{C}$

1

$A + \bar{B} \quad \bar{C} + D$

$+ \quad 1$

1

$A + \bar{C} + 1$

$a + \bar{a}b$

$\bar{D} + \bar{D}\bar{C} = \bar{D} + \bar{C}$

$X + XYZ + \bar{X}Y + WX + \bar{W}X + \bar{X}YZ = X + Y$ $\quad X + Y + Z$ حرام من

$(AB + C + D)(\bar{C} + D)(\bar{C} + D + E) = D + AB\bar{C}$

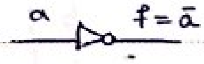
$A\bar{B}C + AB + \overline{ABC} + A\bar{C} + AB\bar{C} = 0$

Scanned by CamScanner

Subject:
Date

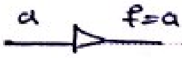
گیت های پایه منطقی

NOT



a	f
0	1
1	0

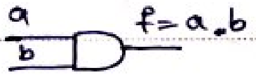
Buffer



a	f
0	0
1	1

سرعت را کمتر می کند. (دیرتر به خط پایانی می رسد)

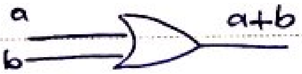
AND



a	b	f
0	0	0
0	1	0
1	0	0
1	1	1

and ← n ورودی در همه حالت ها
0 و در حالتی که همه 1 باشد
1 می دهد.

OR



a	b	f
0	0	0
0	1	1
1	0	1
1	1	1

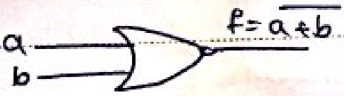
n ورودی ← همیشه 1 مگر زمانی که
تمام ورودی ها 0 باشد که
0 می شود.

NAND

a	b	f
0	0	1
0	1	1
1	0	1
1	1	0

n ورودی ← در همه حالت 1
زمانی که همه 1 شوند ← 0.

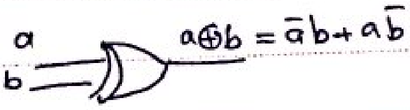
NOR



a	b	f
0	0	1
0	1	0
1	0	0
1	1	0

n ورودی ← فقط وقتی همه
0 باشند 1 می شود.
در باقی حالات 0

XOR



a	b	f
0	0	0
0	1	1
1	0	1
1	1	0

Scanned by CamScanner

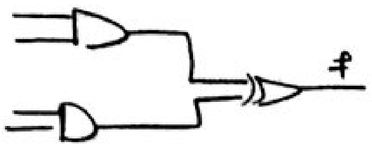
Subject :

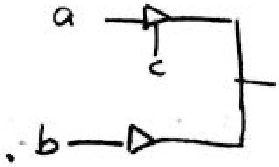
Date

XNOR

$\overline{(a \oplus b)} = a \odot b = \bar{a}\bar{b} + ab$

a	b	f
0	0	1
0	1	0
1	0	0
1	1	1

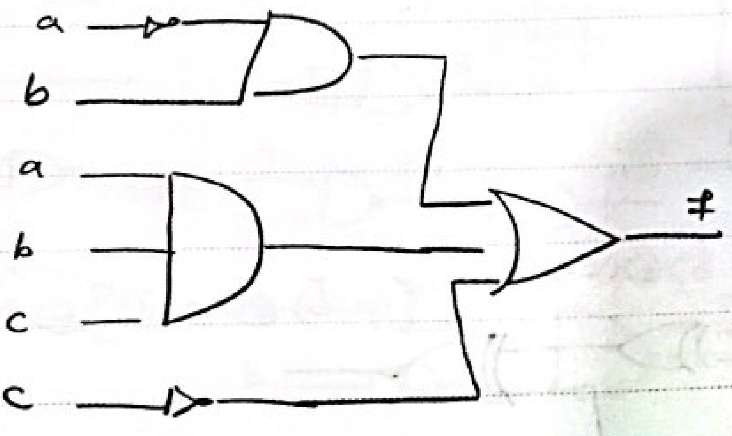




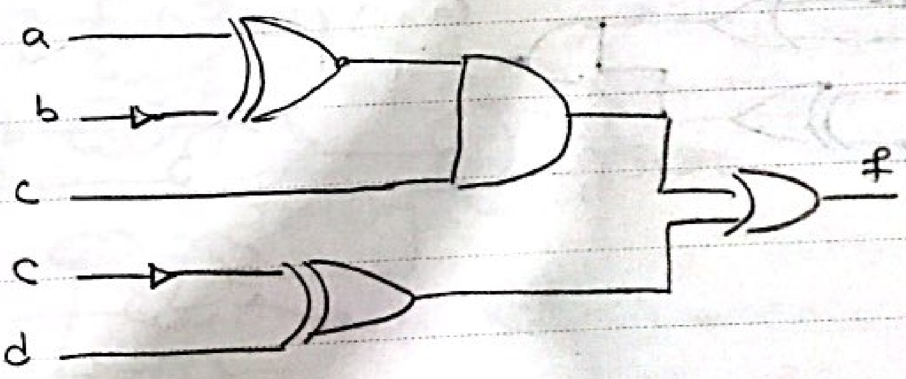
Scanned by CamScanner

Subject. جلسه ۳
Date.

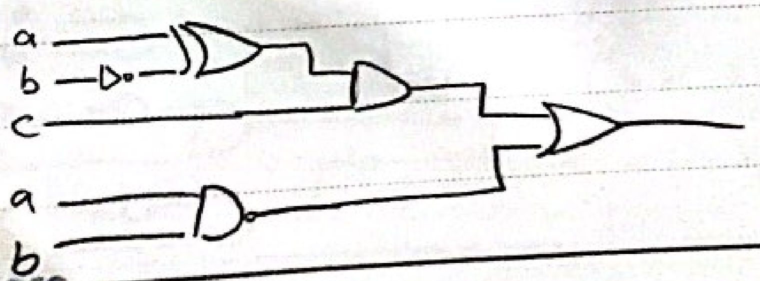
$f(a,b,c) = \bar{a}b + abc + \bar{c}$



$f(a,b,c,d) = (a \odot \bar{b})c + (\bar{c} \cdot \oplus d)$



$f(a,b,c) = (a \oplus \bar{b})c + \overline{ab}$



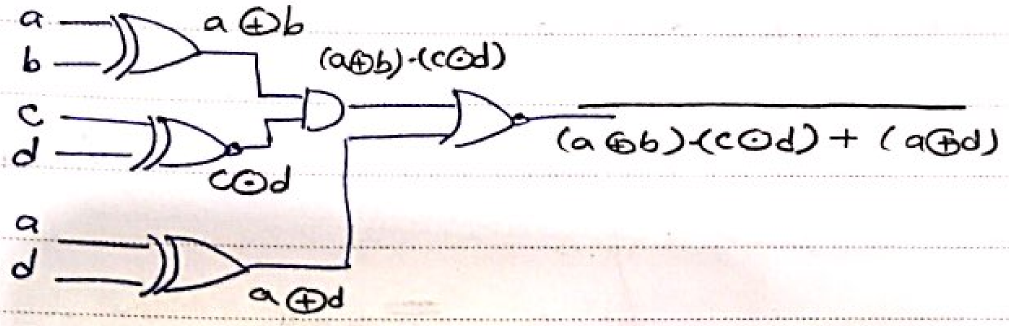
PAPCO

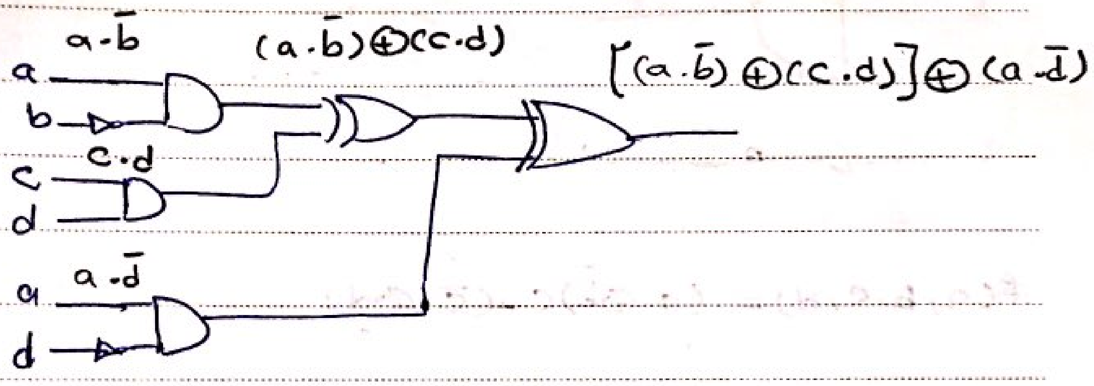
d

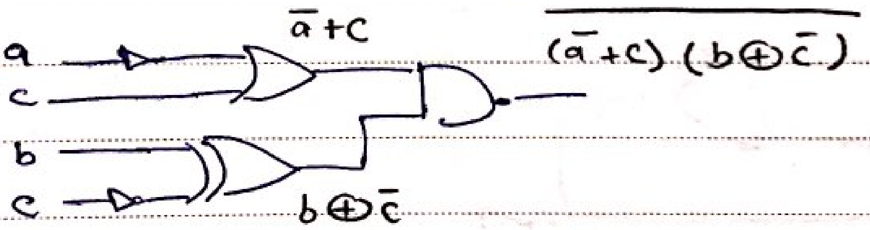
Scanned by CamScanner

Subject.

Date.







Scanned by CamScanner

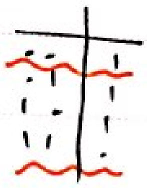
Subject.

Date.

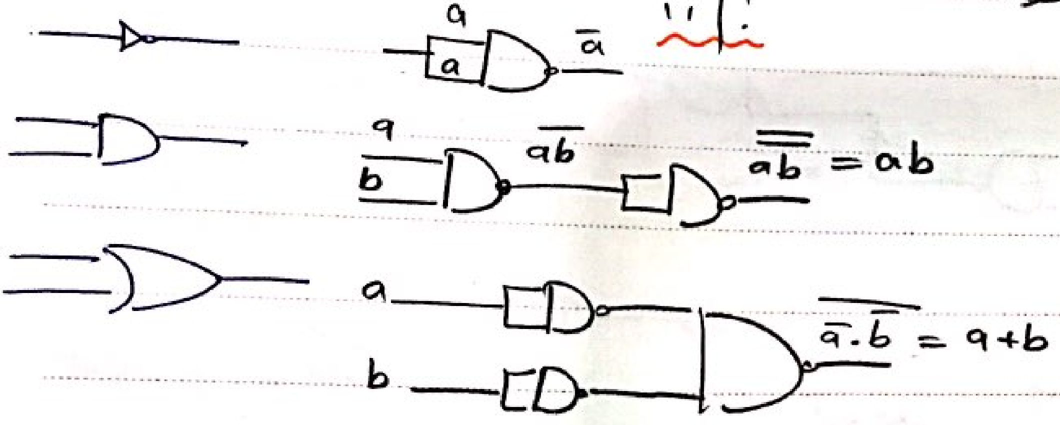
$a \oplus b = \bar{a}b + a\bar{b}$

$a \odot b = \bar{a}\bar{b} + ab$

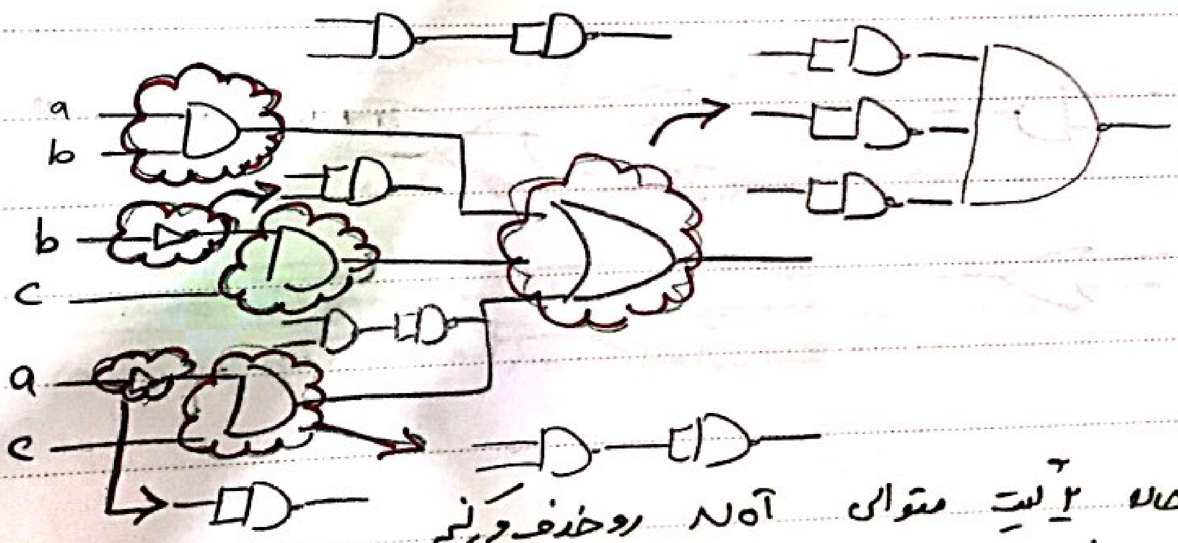
طراحی گیت های universal → NAND, NOR



همیشه ۱ به جز وقتی که ← NAND
هر دو ورودی همزمان ۱ شود

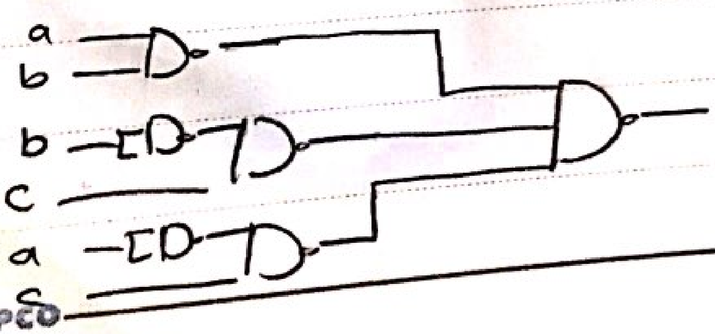


مثال: $f(a,b,c) = ab + \bar{b}c + \bar{a}c$



حالا ۲ گیت متوالی NOT آن رو حذف میکنیم و بجاش سیم می ذاریم.

شکل نهایی

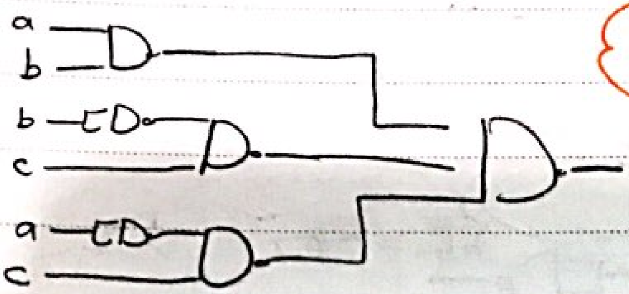


PAPCO

Scanned by CamScanner

Subject.
Date.

راه بعدی ← تبدیل عبارت به NAND فقط SOP های ۲ بار آن NOT میکنیم.

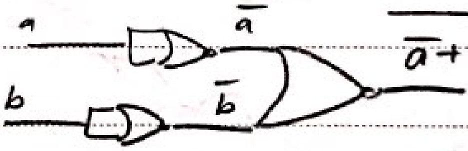


$$\overline{\overline{ab+\bar{b}c+\bar{a}c}} = \overline{\overline{ab}\cdot\overline{\bar{b}c}\cdot\overline{\bar{a}c}}$$

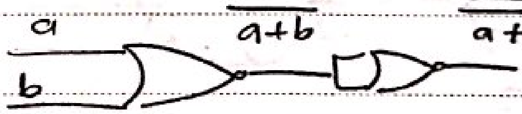
NOR

a	b	
0	0	1
0	1	0
1	0	0
1	1	0

همیشه ۰ مگر وقتی که هردو ۰ باشه ← 1



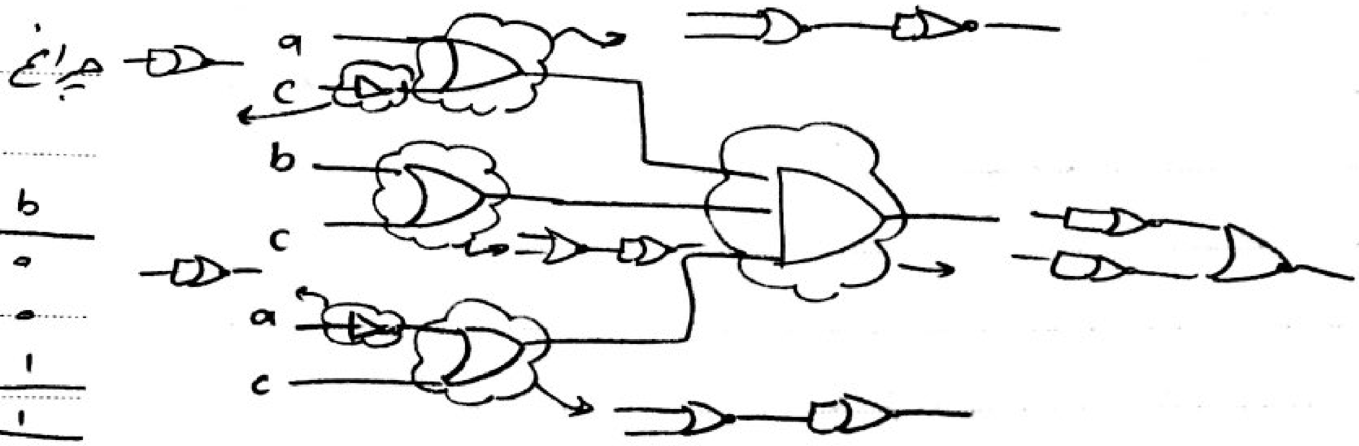
$\overline{\bar{a}+\bar{b}} = ab$



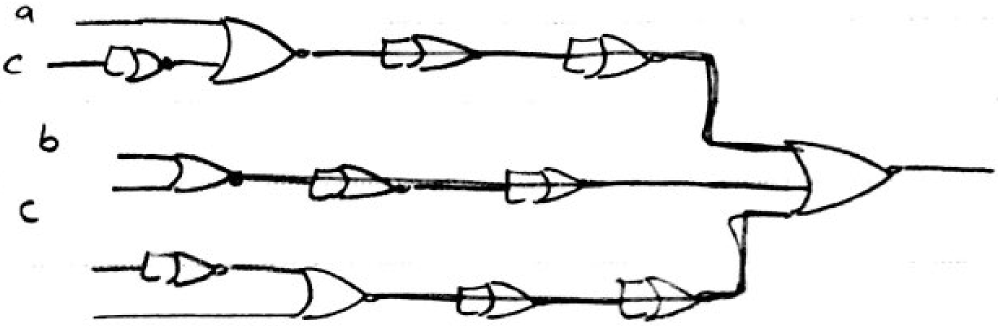
$\overline{\overline{a+b}} = a+b$

Subject.
Date.

$f(a,b,c) = (a+\bar{c})\cdot(b+c)(\bar{a}+c)$



شکل نهایی



راه بعدی ← فرم جمله فقط باید POS باشد، ۲ بار آن NOT می‌کنیم.

$$\overline{\overline{(a+\bar{c})(b+c)(\bar{a}+c)}} = \overline{(a+\bar{c})} + \overline{(b+c)} + \overline{(\bar{a}+c)}$$

Scanned by CamScanner

Subject.
Date.

$ab(c+c') \to abc + abc'$

SoP $\to$ $ab + abc + b\ (a'c' + ac' + a'c + ac) = a'bc' + abc'$

$a'bc + abc$

PoS $\to$ $(a+b)\cdot(a+c)\cdot c$

Minterm (SoP) { $abc' + abc + a'bc' + a'bc$

Maxterm (PoS) }

$f(a,b,c) = (a+b)\cdot(a+c)\cdot c =$

$a+b+c$ $\quad$ $a+b+c$ $\quad$ $\bar{a}+\bar{b}+c$

$a+b+c'$ $\quad$ $a+b'+c$ $\quad$ $\bar{a}+b+c$

$a+\bar{b}+c$

$a+b+c$

Scanned by CamScanner

Subject: جلسه ۶
Date

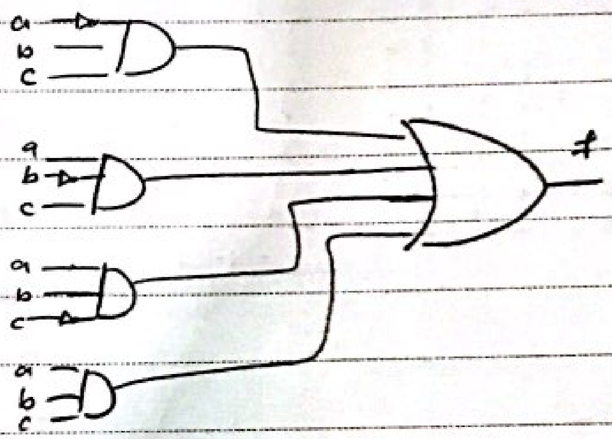
یک مدار طراحی کنید برای رای دادن ۳ نفر تا در صورت حداکثر بودن آرا چراغ مدار روشن شود.

	a	b	c	f
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

1) minterm ⇉ $\bar{a} \to 0$, $a \to 1$ ⇒ بین متغیرها ضرب

$f(a,b,c) = \sum m(3,5,6,7)$

$\bar{a}bc + a\bar{b}c + ab\bar{c} + abc$

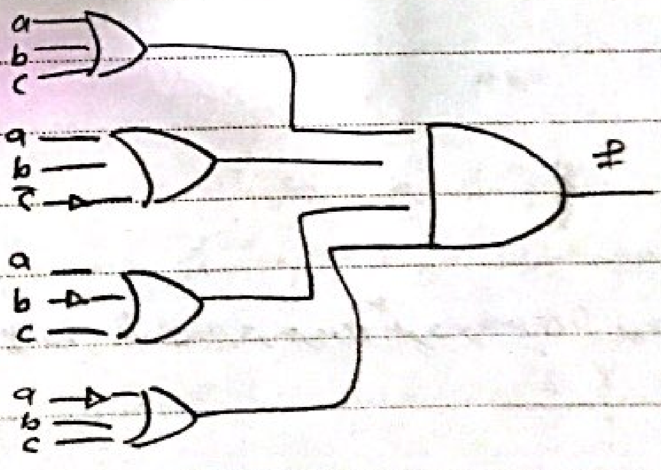


a	b	c	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

2) maxterm ⇉ $0 \to a$, $1 \to \bar{a}$ ⇒ بین متغیرها جمع

$f(a,b,c) = \prod(0,1,2,4)$

$(a+b+c)(a+b+\bar{c})(a+\bar{b}+c)(\bar{a}+b+c)$



Scanned by CamScanner

۵

Subject :
Date

مداری برای ضرب اعداد ۲ بیتی طراحی کنید.

۴ تا ورودی ← ۲ تا ۲ بیتی

۴ بیت ورودی و ۴ بیت خروجی

حالا ۲ بیتی ۲ بیتی جدا می‌کنیم و

ضرب می‌کنیم در خروجی می‌نویسیم

	a_1	a_0	b_1	b_0	f_3	f_2	f_1	f_0
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0
2	0	0	1	0	0	0	0	0
3	0	0	1	1	0	0	0	0
4	0	1	0	0	0	0	0	0
5	0	1	0	1	0	0	0	1
6	0	1	1	0	0	0	1	0
7	0	1	1	1	0	0	1	1
8	1	0	0	0	0	0	0	0
9	1	0	0	1	0	0	1	0
10	1	0	1	0	0	1	0	0
11	1	0	1	1	0	1	1	0
12	1	1	0	0	0	0	0	0
13	1	1	0	1	0	0	1	1
14	1	1	1	0	0	1	1	0
15	1	1	1	1	1	0	0	1

minterm :

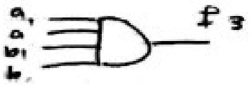
$f_3 = \Sigma m(15) = a_1 a_0 b_1 b_0$

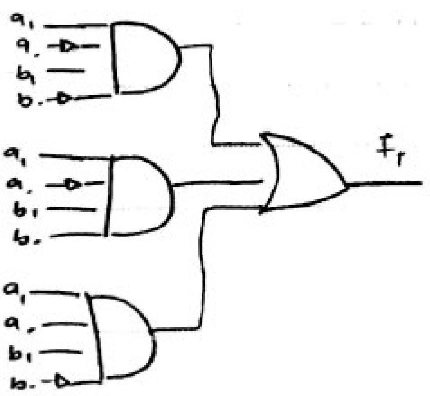
$f_2 = \Sigma m(10, 11, 14) =$

$a_1 \bar{a}_0 b_1 \bar{b}_0 + a_1 \bar{a}_0 b_1 \bar{b}_0 + a_1 a_0 b_1 \bar{b}_0$

$f_1 = \Sigma m(6, 7, 9, 11, 13, 14)$

$f_0 = \Sigma m(5, 7, 13, 15)$





مداری طراحی کنید که عدد ۳ بیتی گری را در ورودی بگیرد و باینری را در خروجی نمایش دهد.

a	b	c		X	Y	Z
0	0	0	0	0	0	0
0	0	1	1	0	0	1
0	1	0	3	0	1	1
0	1	1	2	0	1	0
1	0	0	5	1	0	1
1	0	1	6	1	1	0
1	1	0	4	1	0	0
1	1	1	7	1	1	1

$X = \Sigma m(4, 5, 6, 7)$

PAPCO

Scanned by CamScanner

Subject:
Date

روش های ساده سازی با استفاده از جدول کارنو

n=3 (a,b,c)

a \ bc	00	01	11	10
0	0 (000)	1 (001)	3 (011)	2 (010)
1	4 (100)	5 (101)	7 (111)	6 (110)

n=4 (a,b,c,d)

ab \ cd	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

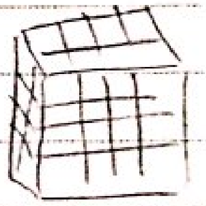
n=5

bc \ de	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

a=0

bc \ de	00	01	11	10
00	16	17	19	18
01	20	21	23	22
11	28	29 (11101)	31	30
10	24	25	27	26

a=1



گروه بندی به توان ۲ ـ بیشترین تعداد ۱ ها ـ ۱ کناری کنار

نوشتن نام گروه به ازای حرفی تغییر کرده باشد نمی نویسیم و مشترک ها رو می نویسیم.

مثال → $f(a,b,c) = \Sigma m(2,3,6)$

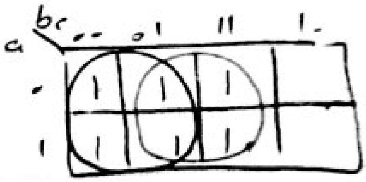
	00	01	11	10
0			1	1
1				1

$\bar{a}b + b\bar{c}$

Scanned by CamScanner

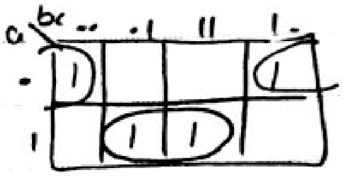
Subject :
Date

$f(a,b,c) = \Sigma m(0,1,3,4,5,7)$



$\bar{b} + c$

$f(a,b,c) = \Sigma m(0,2,5,7)$

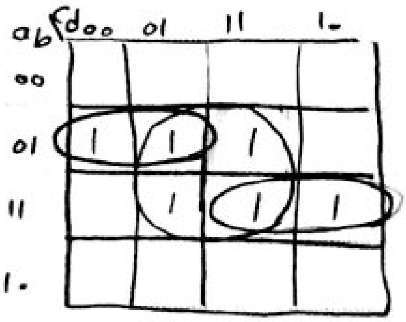


$\bar{a}\bar{c} + ac$

Scanned by CamScanner

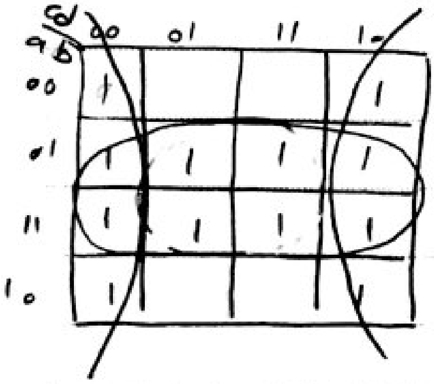
Subject: جلسه ۵
Date

$f(a,b,c,d) = \sum m(4,5,7,13,14,15)$ با استفاده از جدول کارنو ساده کنید.



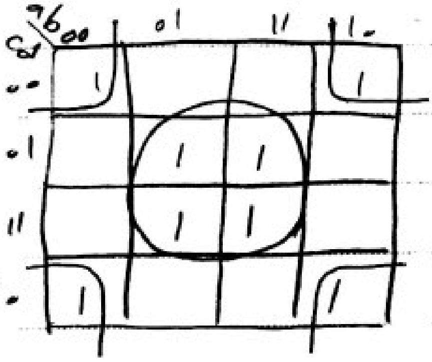
$a'bc' + bd + abc$

$f(a,b,c,d) = \sum m(0,2,4,5,6,7,8,10,12,13,14,15)$



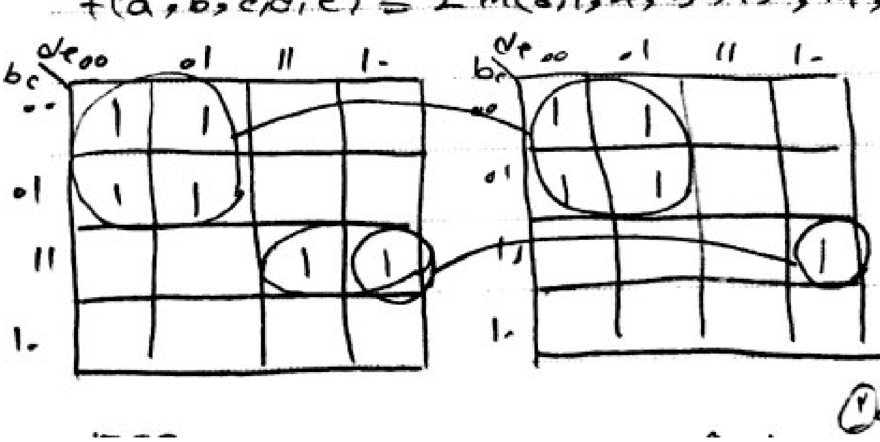
$d' + b$

$f(a,b,c,d) = \sum(0,2,5,7,8,10,13,15)$



$bd + \bar{b}\bar{d}$

$f(a,b,c,d,e) = \sum m(0,1,4,5,15,14,16,17,20,21,30)$



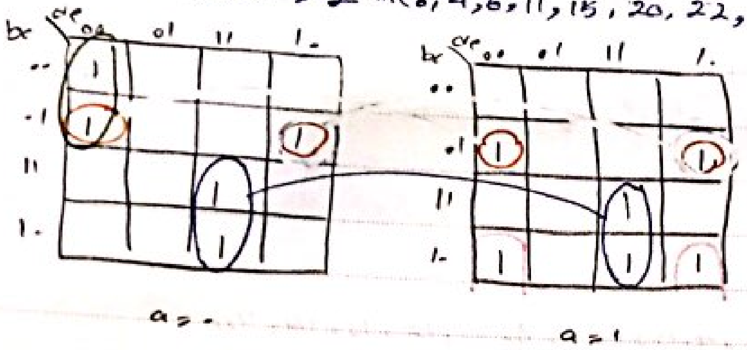
$\bar{b}\bar{d} + \bar{a}bcd + bcd\bar{e}$

در هر تغییره می‌بینیم که دسته چندتایی داریم و ۲ به توان چه توری است مثلا
اگر دسته ۴تایی داشتیم بالتیم یعنی ۲ به توان ۲

Scanned by CamScanner

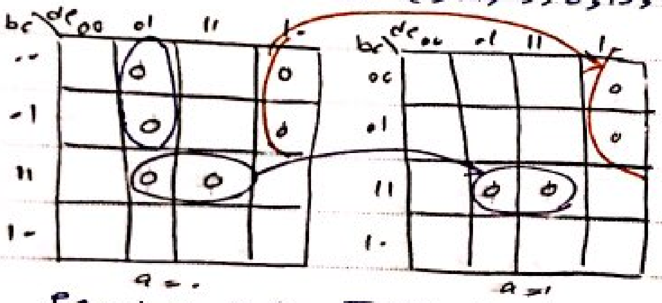
Subject:
Date

$f(a,b,c,d,e) = \sum m(0,4,6,11,15,20,22,24,26,27,31)$



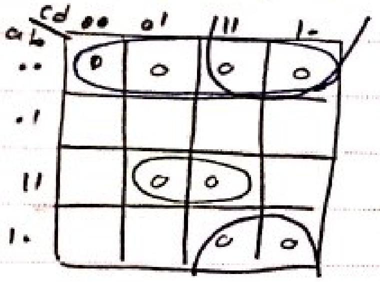
$\bar a \bar b \bar d \bar e + \bar b c \bar e + bde + ab\bar c \bar e$

$f(a,b,c,d,e) = \prod M(0,2,5,6,13,15,18,22,29,31)$



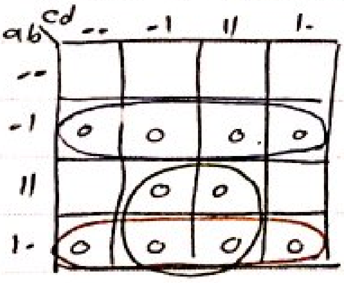
$(b+\bar d+e)(a+b+d+\bar e)(\bar b+\bar c+\bar e)$

$f(a,b,c,d) = \prod M(0,1,2,3,10,11,13,15)$



$(a+b)(b+\bar c)(\bar a+\bar b+\bar d)$

$f(a,b,c,d) = \prod M(4,5,6,7,8,9,10,11,13,15)$

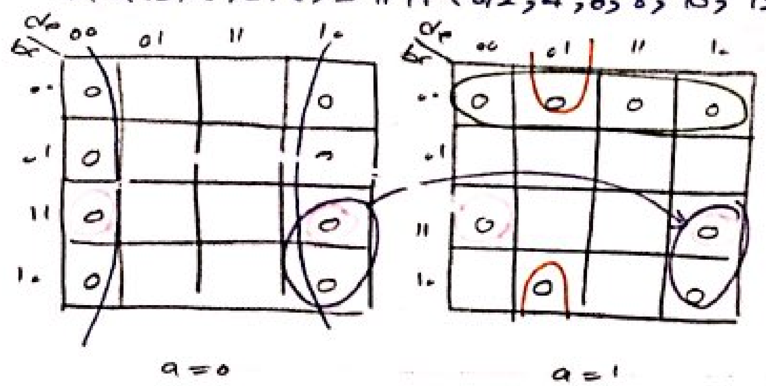


$(a+\bar b)(\bar a+b)(\bar a+\bar d)$

PAPCO

Subject
Date

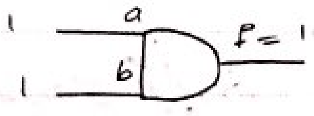
$f(a,b,c,d,e) = \prod M(0,2,4,6,8,10,12,14,16,17,18,19,25,26,28,30)$



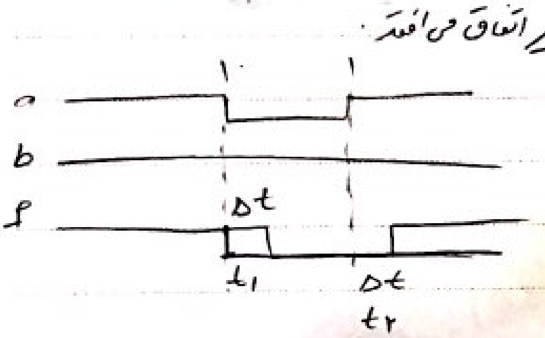
$(a+e)(\bar{b}+\bar{c}+e)(\bar{b}+\bar{d}+e)$

$(\bar{a}+b+c)(\bar{a}+c+d+\bar{e})$

جلسه ۶

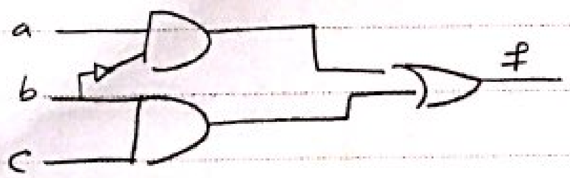


تاخیر گیت‌ها: هنگامی که a و b هر دو ۱ باشن خروجی ۱ می‌شود
ولی زمانی که مثلا a ← ۰ می‌شود اگر گیت ایده‌آل باشد بلافاصله
۰ می‌شد ولی در حقیقت بعد از یک مدت زمانی (تاخیر) این تغییر اتفاق می‌افتد.
این تاخیرهای زمانی با هم برابر نیستند

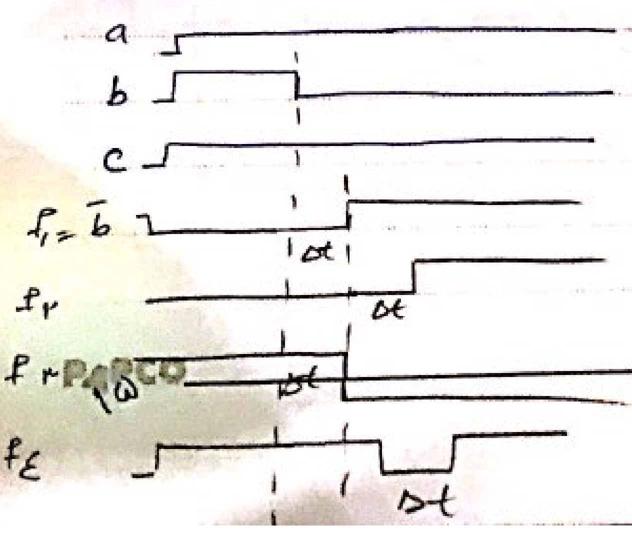


$\Delta t = \frac{t_1 + t_2}{2}$

EX:



همه گیت‌ها تاخیر Δt دارند.



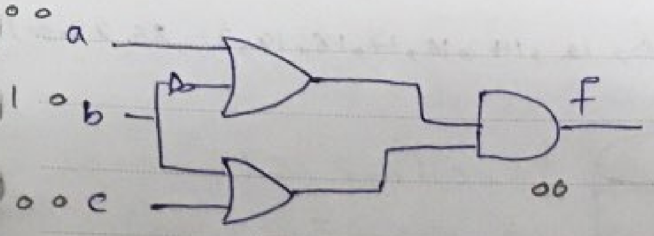
راه حل: با بافر تاخیرهای زمانی را برابر کنیم.

Hazard:

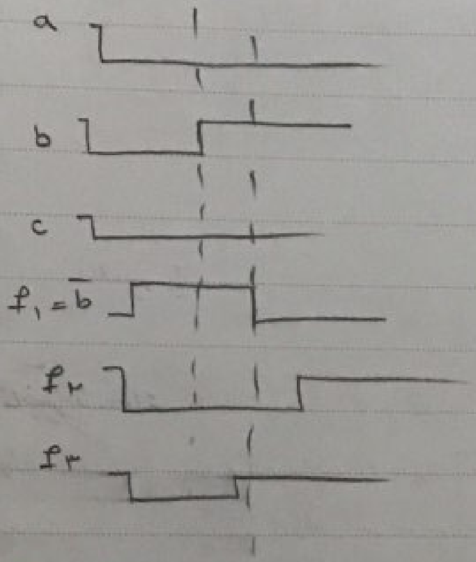
نکته: قطعه‌ای از ورودی تغییر می‌دهد که در مسیرهای مختلف تاخیرهای مختلف

Subject
Date

تمرین:

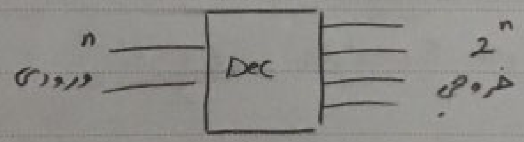


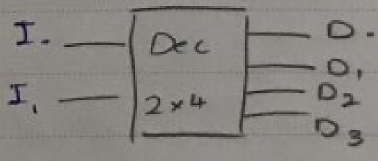
?



مدارهای ترکیبی

Decoder → ماژول هایی با n پایه ورودی و 2^n پایه خروجی





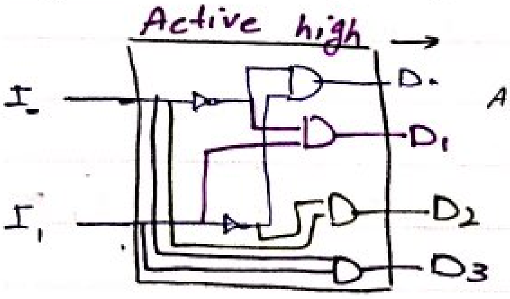
I_0	I_1	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

در هر حالت با تغییر ورودی خروجی فعال تغییر میکند.

Subject
Date

$D_0 = \bar{I}_0 \bar{I}_1$ $D_1 = \bar{I}_0 I_1$ برای رسم ← مین ترم آن ها را می نویسیم

$D_2 = I_0 \bar{I}_1$ $D_3 = I_0 I_1$



با یه ۰ فعال می شد

Active high ⇒ ۰ ← غیر فعال فعال بودن ← ۱

Active low ⇒ ۱ ← غیر فعال فعال ← ۰

دکدر $n \times 2^n$ چه خروجی و چه تعداد از آن ها دارد؟ 2^n AND n ورودی

n NOT

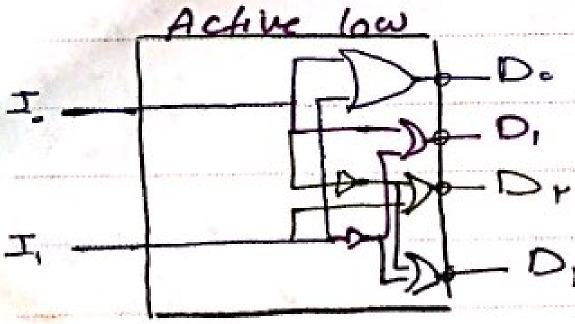
دکدری که در منطق low باشد در شکل شماتیک قبل هر خروجی با bubble ظاهر می شد.

در مثال قبلی →

I_0	I_1	D_0	D_1	D_2	D_3
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

→ بهتره از maxterm استفاده کنیم. چون اینجا تعداد ۰ ها کمتره.

$D_0 = I_0 + I_1$ $D_1 = I_0 + \bar{I}_1$ $D_2 = \bar{I}_0 + I_1$ $D_3 = \bar{I}_0 + \bar{I}_1$



۴ NAND n ورودی

n NOT

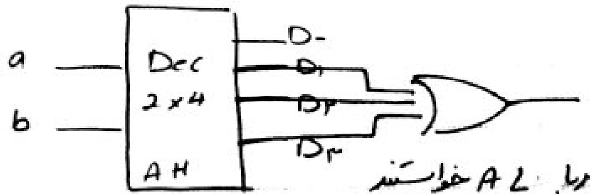
Subject
Date

از روی مین‌ترم بدست بیاریم

$f(a,b) = \sum m(1,2,3) \rightarrow \bar{a}b + a\bar{b} + ab$

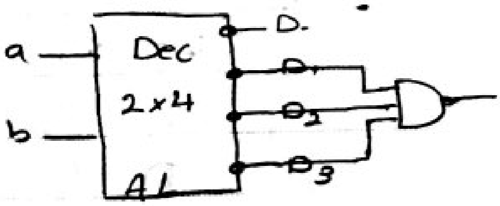
با دیکودر رسم کنید.

روی این بیت‌ها رسدها عبارت مین‌ترم‌ها را بدست می‌آوریم و OR می‌کنیم.

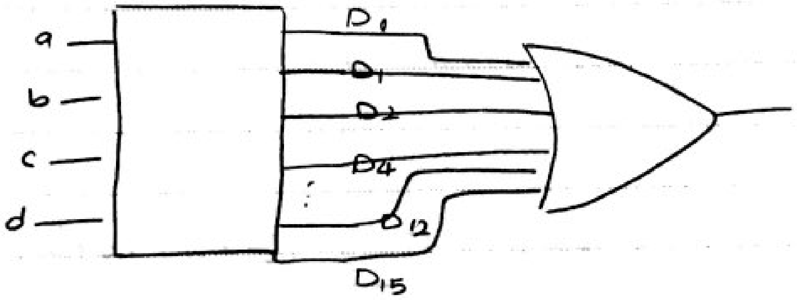


اگر با AL خواستند

OR ای که تمام ورودی‌های آن NOT شده باشد ← NAND



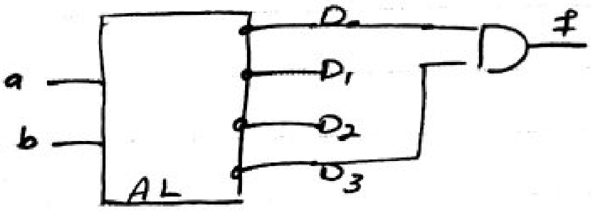
$f(a,b,c,d) = \sum m(0,1,2,4,12,15)$



Maxterm

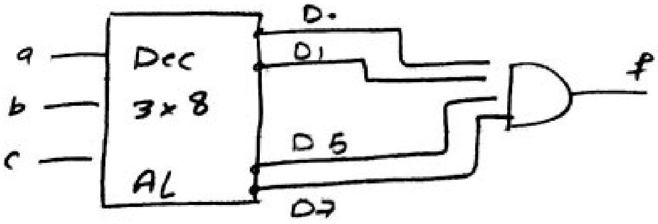
$f(a,b) = \prod M(0,3)$

راحت‌تر با AL کار کنیم



$f(a,b,c) = \prod M(0,1,5,7)$

در این جا (ماکسترم) اگر بخواهیم با AH بسازیم باید با گیت NOR رسم کنیم.



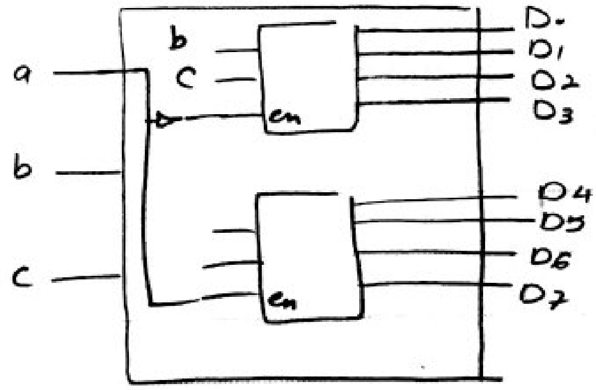
Subject
Date

با استفاده از دیکودر 4×16 مداری برای ضرب اعداد ۲ بیتی طراحی کنید.

0	0	0	0
0	0	0	1
		1	0
		1	1
		0	0
		0	1
		1	0
		1	1

ساخت دیکودرهای بزرگتر به پایه enable دارد که باعث میشه بین دیکودرها انتخاب کنیم.

طراحی دیکودر 3×8 با استفاده از 2×4



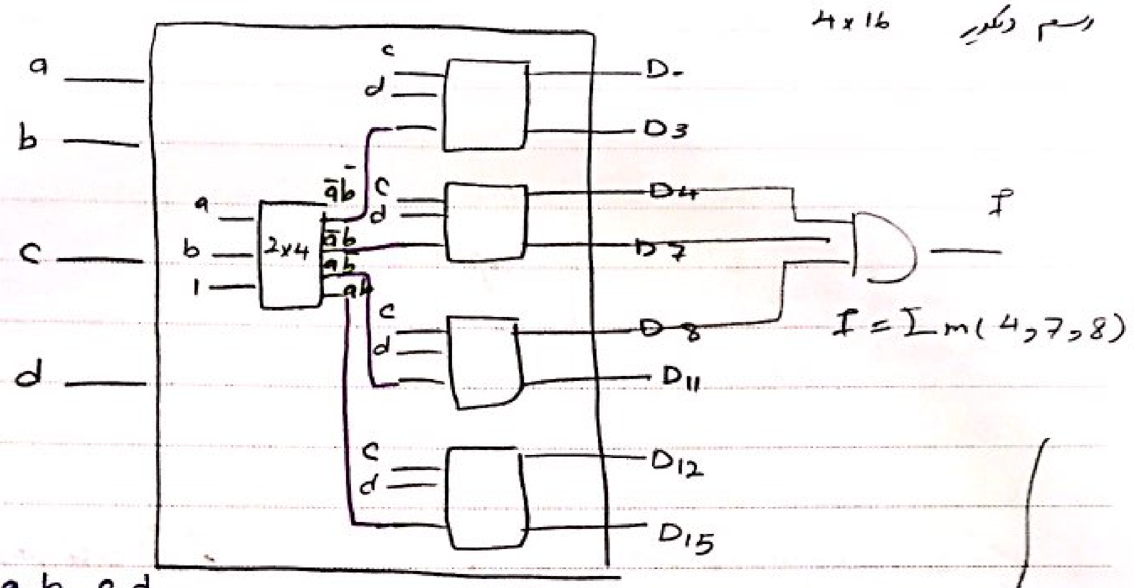
مربوط به enable

a (enable)	b	c	D0	D1	D2	D3	D4	D5	D6	D7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

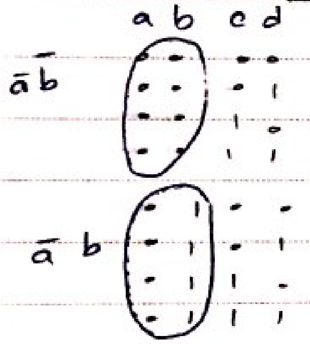
Scanned by CamScanner

Subject
Date

رسم دکودر 4x16



$F = \Sigma m(4, 7, 8)$

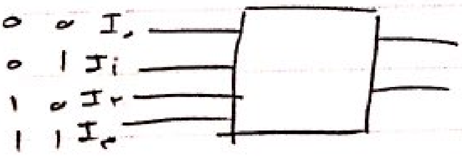


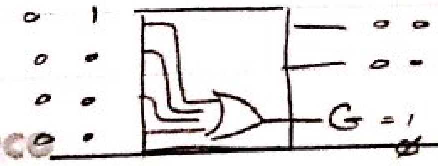
این‌ها برای سخت کردن با enable ها بازی می‌کنیم و جاشونو عوض می‌کنه

$a\bar{b}$ 10

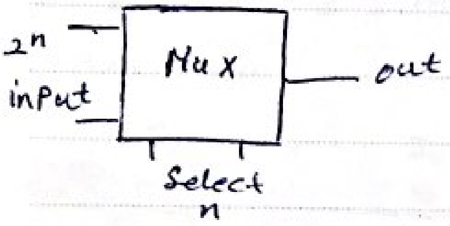
ab 11

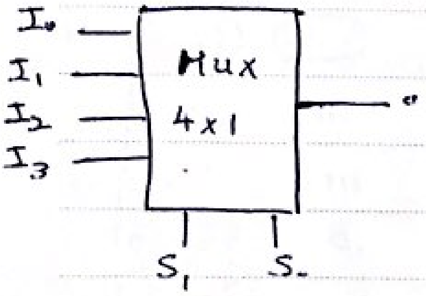
encoder 2^n ورودی n خروجی . اگر همزمان چند ورودی ۱ بود اونی که اولویت بالاتری داشته باشه در خروجی نمایش داده می‌شه.





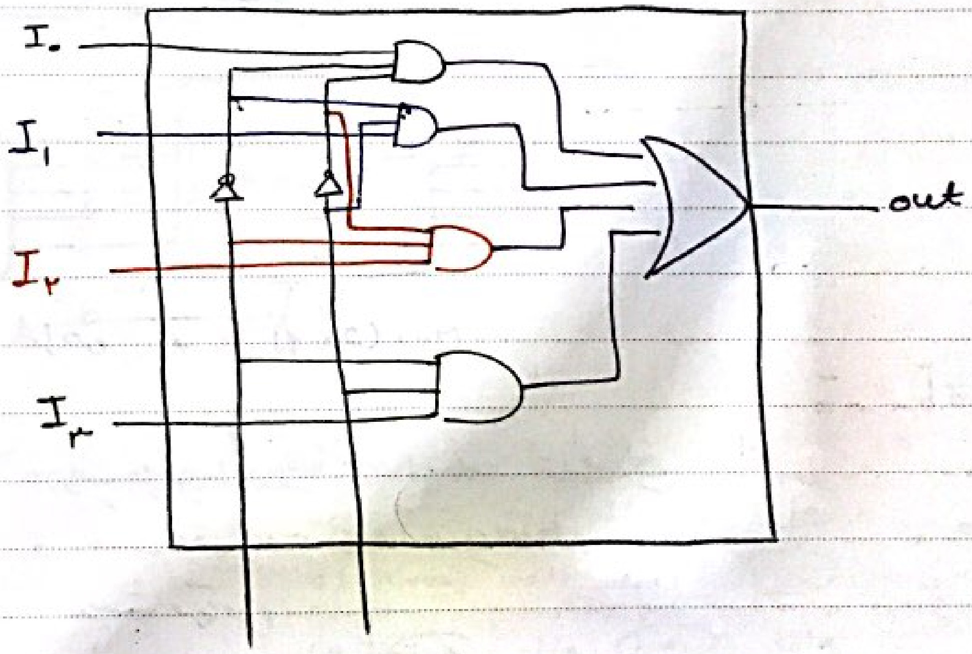
Mux (حالت کلی)





S_1	S_0	f
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

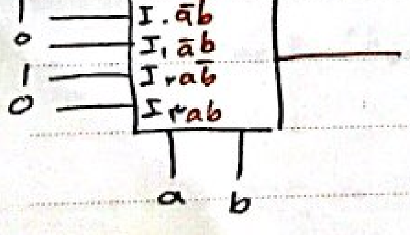
$$f = \bar{S}_1\bar{S}_0 . I_0 + \bar{S}_1 S_0 . I_1 + S_1\bar{S}_0 . I_2 + S_1 S_0 . I_3$$



*

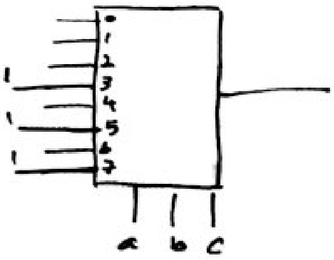
$$f(a,b) = \Sigma m(0,2) = \bar{a}\bar{b} + a\bar{b}$$

از روی جدول selector ها تقسیم می‌کنیم و مقادیر I ها از روی این انتخاب می‌کنیم که اگر مقدار I یک باشد جمله وجود دارد و اگر صفر باشد جمله وجود ندارد.

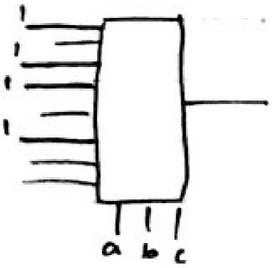


Scanned by CamScanner

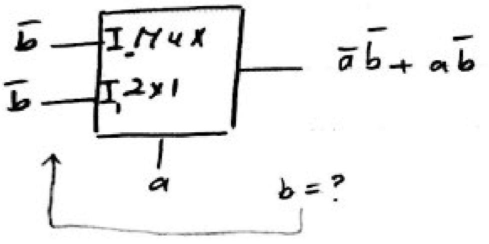
$f(a,b,c) = \Sigma m(3,5,7)$



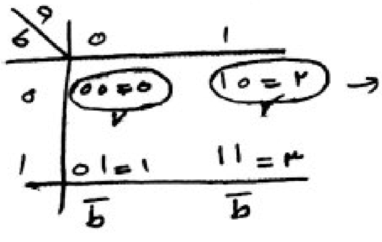
$f(a,b,c) = \Pi M(1,4,6,7) = \Sigma m(0,2,3,5)$



طراحی * با Mux (2x1)



در این جا یکی را در پایه Selecter قرار می دهیم
و حرف باقی مانده را از طریق جدول
بدست می آوریم.



چون در صورت سوال Σ(0,1) بود ما خانه های 0
سطر اول را انتخاب می کنیم که همان $\bar{b}$ می شود.

Scanned by CamScanner

Subject
Date

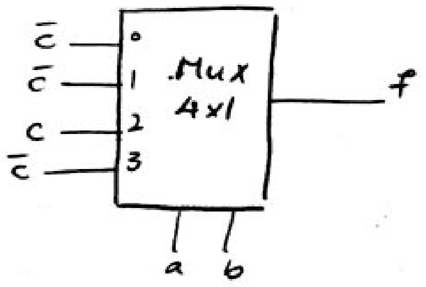
$f(a,b,c,d) = \Sigma m(0,1,4,5,10,11,12,13)$

Mux 4x1

cd \ ab	I_0 00	I_1 01	I_2 10	I_3 11
00	(0000)	(0100)	1000	(1100)
01	(0001)	(0101)	1001	(1101)
10	0010	0110	(1010)	1110
11	0011	0111	(1011)	1111
	$\bar{c}\bar{d}+\bar{c}d$	$\bar{c}$	$c\bar{d}+cd$	$\bar{c}$

$\bar{c}\bar{d}+\bar{c}d = \bar{c}(d+\bar{d})$, $d+\bar{d} = 1$; $c\bar{d}+cd = c$

→ فقط به cd کار داریم.
یعنی متغیرهای باقی مانده



محل ها a,c آدرس →

bd \ ac	00	01	10	11
00	(0000)	0010	1000	(1010)
01	(0001)	0011	1001	(1011)
10	(0100)	0110	(1100)	1110
11	(0101)	0111	(1101)	1111
	1	0	$b\bar{d}+bd$	$\bar{b}$

چون تمامی ستون ها انتخاب شده ← 1

0 ↓ چون هیچ ستونی انتخاب نشده.

$b\bar{d}+bd = b$

Scanned by CamScanner

Subject

Date

$f(a,b,c,d) = \prod M\,(1,2,5,6,7,12,15)$

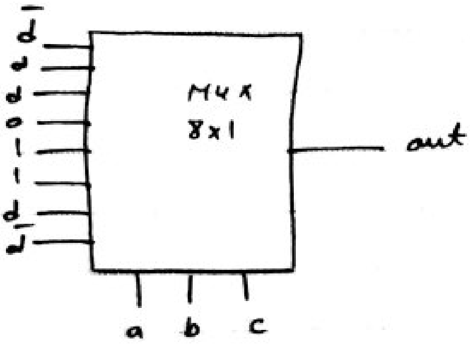
Mux
8x1

d \ abc	000	001	010	011	100	101	110	111
0	0000	(0010)	0100	(0110)	1000	1010	(1100)	1110
1	(0001)	0011	0101	(0111)	1001	1011	1101	(1111)
	$\bar{d}$	d	$\bar{d}$	0	1	1	d	$\bar{d}$

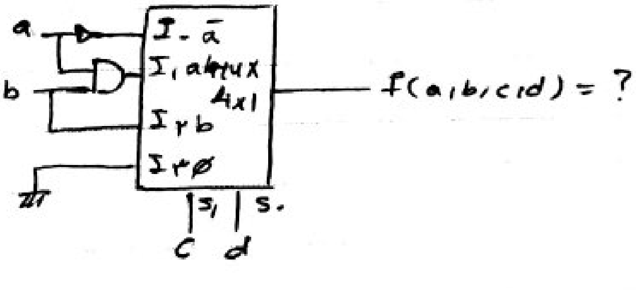
درست برعکس مین ترم.

در عبارت ماکسترمی آنرا صفر با ۱ و ۰ نشون می دادیم.

((معمولا شرح مسئله سوال میشن))



Ex:



$f(a,b,c,d) = ?$

از روی → $\bar{S}_1 \bar{S}_0 I_0 + \bar{S}_1 S_0 I_1 + S_1 \bar{S}_0 I_2 + S_1 S_0 I_3$

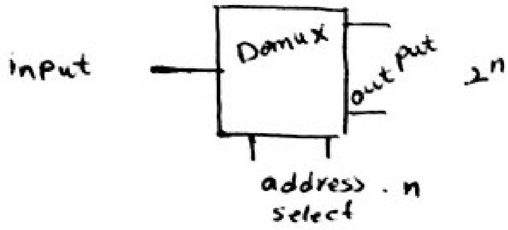
$= \bar{a}\bar{c}\bar{d} + ab\bar{c}d + bc\bar{d}$

$0100 = 4 \leftarrow$

چون مین ترم است باید تمام حروف را داشته باشد. باید حوزش و ۱ و ۰ اش را بذاریم.

حالا ۱ و ۰ می داریم و مین ترم را بدست می آوریم.

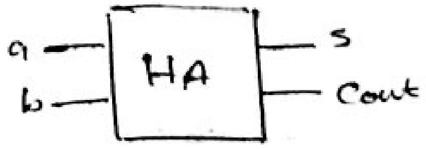
$f(a,b,c,d) = \sum m(0,4,6,13,14)$



در حالت یکسر ۱ ورودی 2^n خروجی.

Adder → half ✓
↳ full

جمع کننده ها (آدر ها)

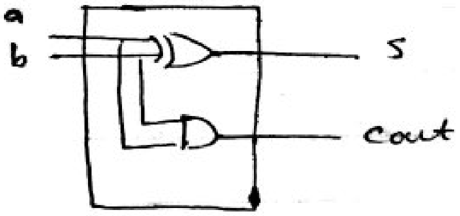


a	b	S	Cout
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

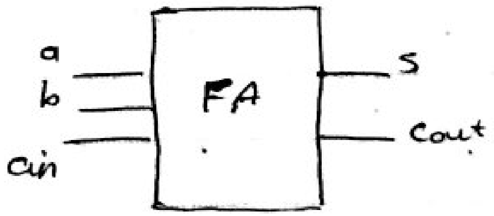
$S = \Sigma m(1,2) = \bar{a}b + a\bar{b} = a \oplus b$

$C_{out} = \Sigma m(3) = ab$

carry ورودی در نیم جمع کننده لحاظ نشده.



full Adder →



a	b	cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$S = \Sigma m(1,2,4,7)$

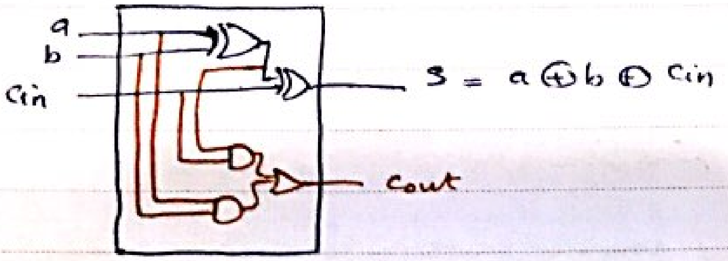
$\bar{a}\bar{b}\,c_{in} + \bar{a}b\,\bar{c}_{in} + a\bar{b}\,\bar{c}_{in} + ab\,c_{in} = a \oplus b \oplus c$

$C_{out} = \Sigma m(3,5,6,7) = \bar{a}b\,c_{in} + a\bar{b}\,c_{in} + ab\bar{c}_{in} + ab\,c_{in}$

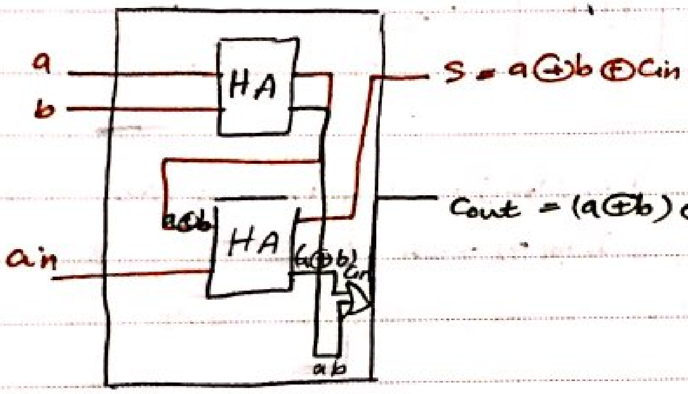
$(a \oplus b)\,c_{in} + ab$

Scanned by CamScanner

Subject
Date



Ex : با یک نیم جمع کننده مدار تمام جمع کننده بسازید.



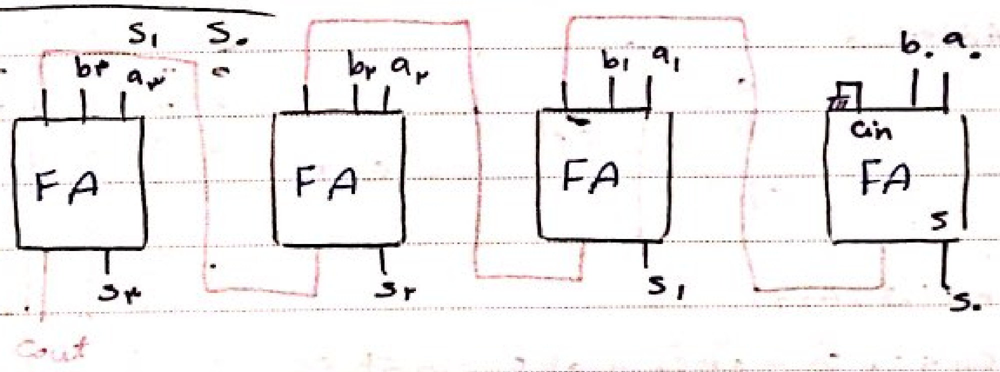
Full Adder ۴ بیتی: با ۴ بیتِ ورودی نیاز به ۴ FullAdder داریم.

C_1 C_0

$A = a_3 \; a_2 \; a_1 \; a_0$

$B = b_3 \; b_2 \; b_1 \; b_0$

$S_1 \; S_0$



Subject

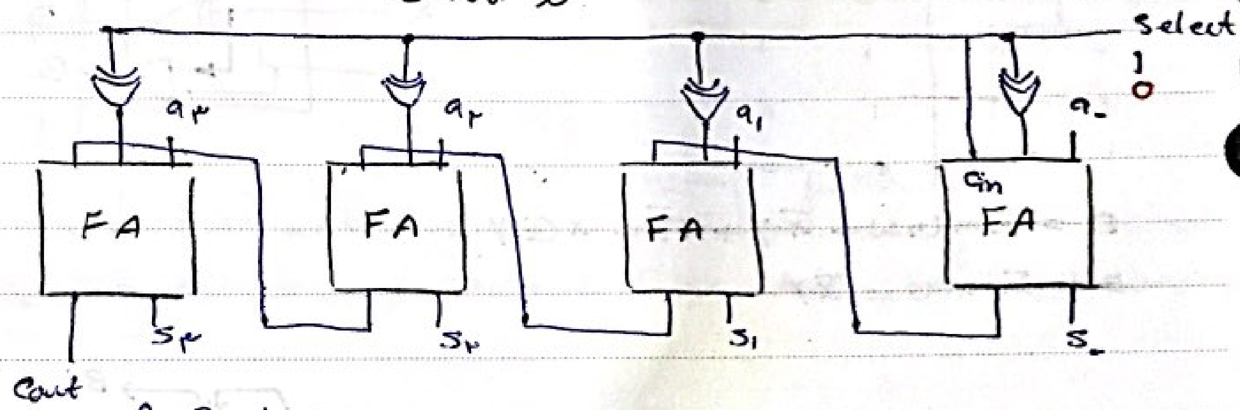
Date

مداری طراحی کنید که بتوان هم A+B و هم A-B را نشان دهد (۴ بیتی)

		XOR
0	0	0
0	1	1
1	0	1
1	1	0

هرچی با ۰ XOR شده خودش می‌شه.

و هرچی با ۱ XOR شده شده آن NOT اش.



A+B+1

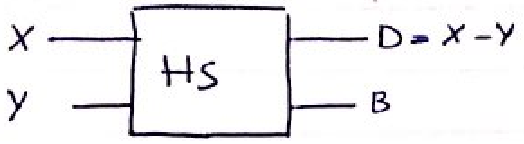
A+B

فقط با ورودی های دوم سروکار داریم.

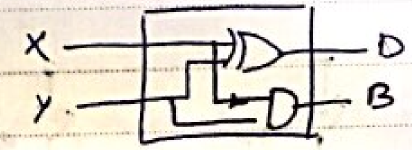
Subject

Date

تفریق کننده { ناقص / کامل

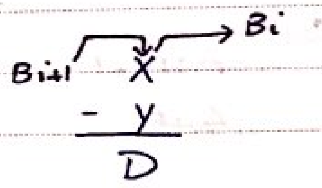


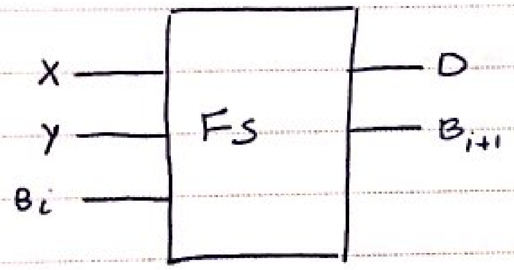
X	Y	D	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

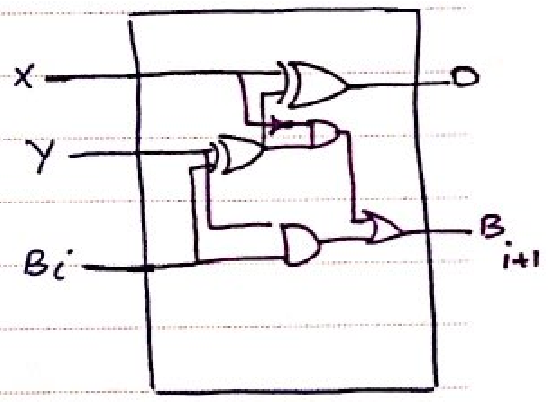


$D = \Sigma m(1,2) = \bar{x}y + x\bar{y} = x \oplus y$

$B = \Sigma m(1) = \bar{x}y$







X	y	Bi	D	Bi+1
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

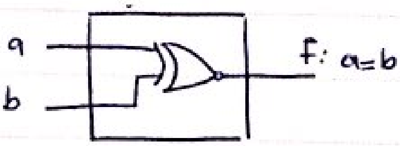
$D = \Sigma m(1,2,4,7) = \bar{x}\bar{y}B_i + \bar{x}y\bar{B_i} + x\bar{y}\bar{B_i} + xyB_i = X \oplus Y \oplus B_i$

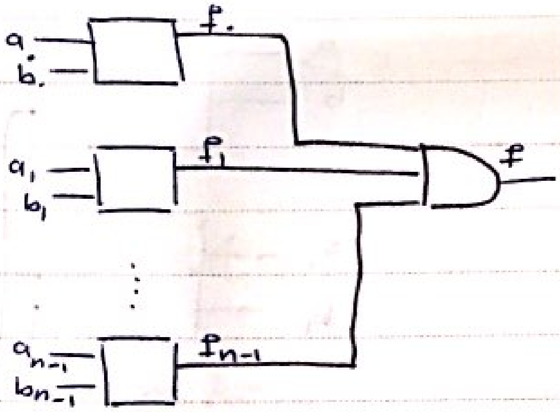
$B_{i+1} = \Sigma m(1,2,3,7) = \bar{x}\bar{y}B_i + \bar{x}y\bar{B_i} + \bar{x}yB_i + xyB_i = \bar{x}(y \oplus B_i) + yB_i$

Scanned by CamScanner

Subject
Date

مقایسه کننده
- ناقص
- کامل



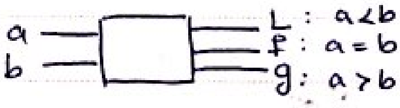


a	b	f
0	0	1
0	1	0
1	0	0
1	1	1

$$f = \Sigma m(0,3) = \bar{a}\bar{b} + ab = a \odot b$$

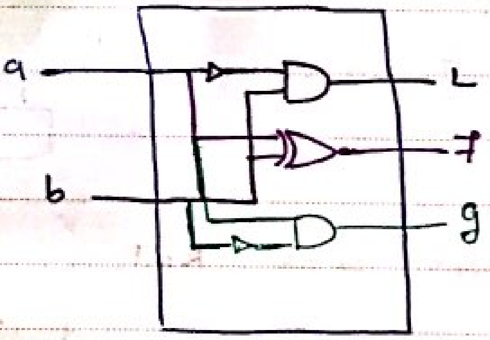
«فقط از گیت AND برای آن که همه باهم ۱ شود پس از این فقط می‌شه استفاده کرد»

کامل ←



→ بیت با ارزش نشان بزرگتر باشد

a	b	L	f	g
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0



$L = \bar{a}b$

$f = a \odot b$

$g = a\bar{b}$

به تعداد بیت‌ها مقایسه کننده احتیاج داریم.

Scanned by CamScanner

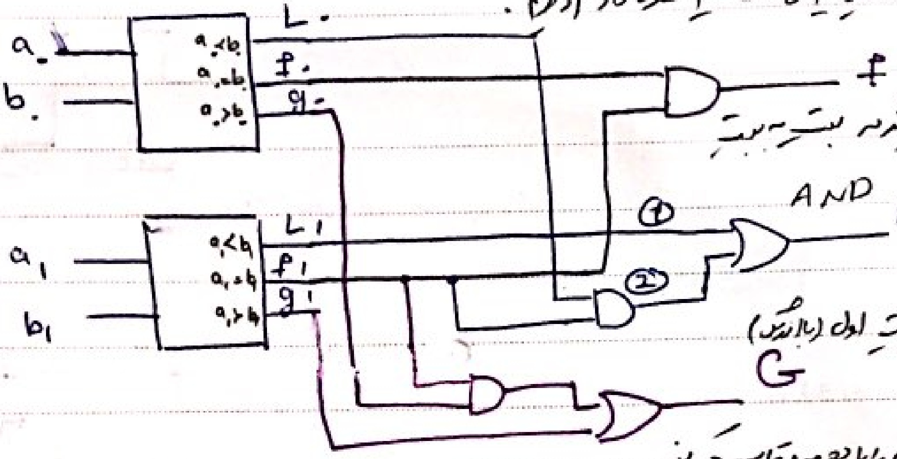
Subject
Date

۴۸ : مقایسه کننده ای طراحی کنید که اعداد ۲ بیتی را مقایسه کند.

$A = a_1 a_0$

$B = b_1 b_0$

به تعداد بیت هایمان ۲ تا مقایسه کننده لازم داریم.



وقتی با هم برابر اند هر بیت به بیت با هم برابر باشه AND

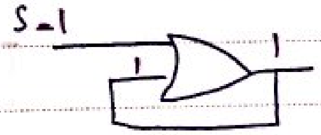
بزرگتر = ① بیت اول (بالایی) بزرگتر باشد.

② رقم بعدی را با هم مقایسه می کنیم.

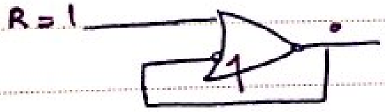
کوچکتره ① $a_1 < b_1$ ② $a_1 = b_1$ ← بعد a_0 و b_0 رو تست می کنیم

مدارهای ترتیبی

Set Latch →

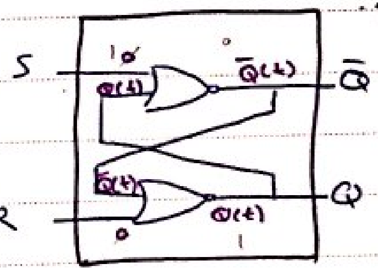


Reset latch



مدار می نیاز داریم که همه کار رو باهم انجام بده.

SR latch →



خروجی لحظه قبل ورودی بعدی می شه.

اول ثابت شد موقعیت با حساب داریم

بعد خروجی ها رو کم می کنیم

سپس وضعیت stable حافظه

اگر همواره ورودی های پایه یکسان توجه به ورودی بعدی خروجی ۱ می شه. OR

Subject

Date

NOR →

00	1
01	0
10	0
11	0

S	R	$Q(t+1)$	$\bar{Q}(t+1)$
0	0	$Q(t)$	$\bar{Q}(t)$
0	1	0	1
1	0	1	0
1	1	X	X

X → وضعیت نامشخص

t لحظه قبل   t+1 لحظه حال

جدول مشخصه →

S	R	$Q(t+1)$
0	0	$Q(t)$
0	1	0
1	0	1
1	1	X

معادله حالت

$$Q(t+1) = S + \bar{R}Q(t)$$

S	R	$Q(t)$	$Q(t+1)$
0	0	0	0 ✓
0	0	1	1
0	1	0	0 ✓
0	1	1	0 ✓
1	0	0	1 ✓
1	0	1	1
1	1	0	X
1	1	1	X

جدول حالت

S \ RQ				
0		1		
1	1	1	X	X

جدول تحریک

↓

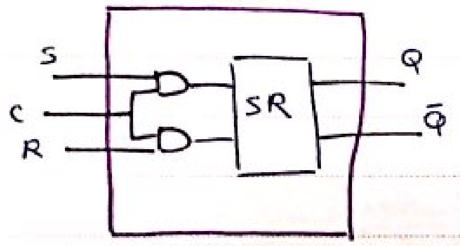
از روی جدول حالت

$Q(t)$	$Q(t+1)$	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

مشکل SR ← ورودی ها به شدت تحت تاثیر سطح اند.

Scanned by CamScanner

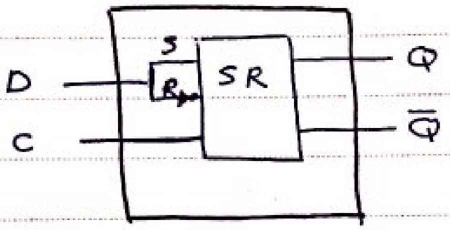
Subject
Date



flip flop : ساده‌ترین راه حل مشکل SR latch

یک پایه clock اضافه می‌کنیم هر وقت c صفر باشد حفظ حالت داریم اگر ۱ باشد جدول مشخصه آن مثل SR latch خواهد بود.

D_ flip flop



D	Q(t+1)
0	0
1	1

جدول مشخصه

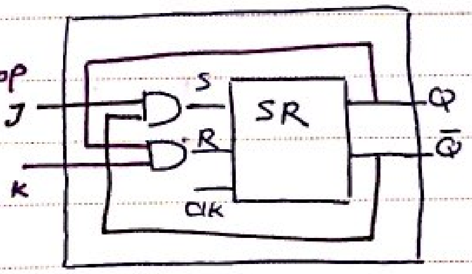
$Q(t+1) = D$

معادله مشخصه

Q(t)	Q(t+1)	D
0	0	0
0	1	1
1	0	0
1	1	1

جدول تحریک

JK flip flop



J	K	Q(t)	Q(t+1)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

J	K	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	$\bar{Q}(t)$

← جدول حالت *

	1		
1	1		1

$Q(t+1) = \bar{K}Q(t) + J\bar{Q}(t)$

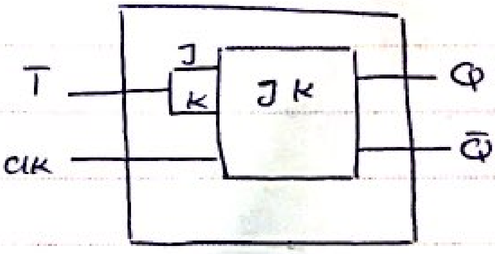
Subject
Date

Q(t)	Q(t+1)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

جدول تحریک

از روی * بدست می‌آوریم .

T flip flop



T	Q(t+1)
0	Q(t)
1	$\bar{Q}(t)$

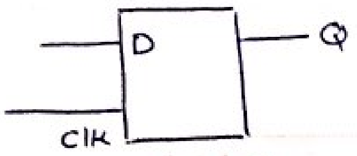
$$Q(t+1) = \bar{T}Q(t) + T\bar{Q}(t) = T \oplus Q(t)$$

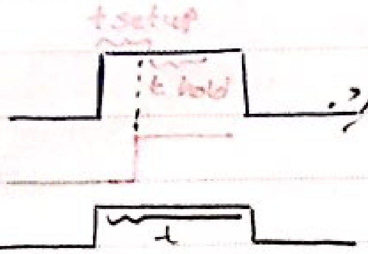
Q(t)	Q(t+1)	T
0	0	0
0	1	1
1	0	1
1	1	0

Scanned by CamScanner

Subject
Date

فلیپ فلاپ ها حساس به سطح هستند.



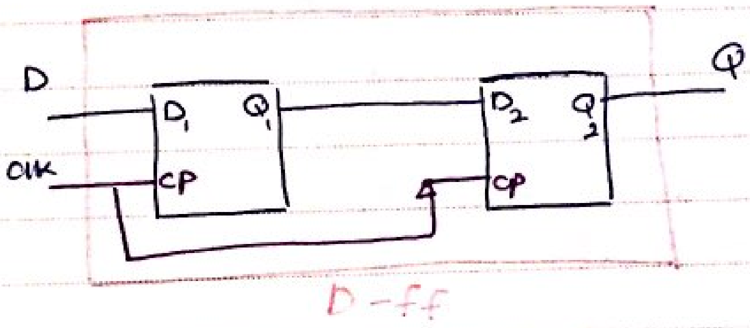


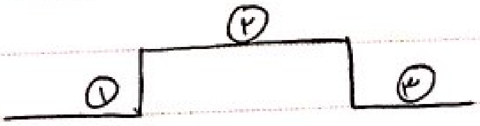
t setup ← ورودی فلیپ فلاپ باید یک مقدار قبل از پالس کلاک مقدارش را بگیرد.

t hold ← وقتی مقدار پایدار در آن حالت بمونه (بعد از پالس)

طراحی فلیپ فلاپی که حساس به لبه باشد:

Master_Slave





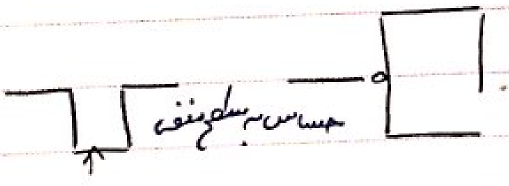
حساس به لبه پایین رونده.

۱و۲ D_1: روشن Q_1 ⊓⊓⊓

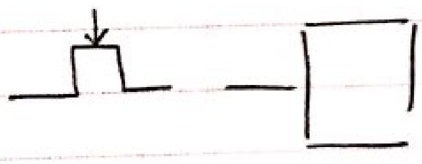
D_2: خاموش Q: حفظ مقدار قبلی

۳ D_1: خاموش Q_1: حفظ آخرین مقدار قبلی

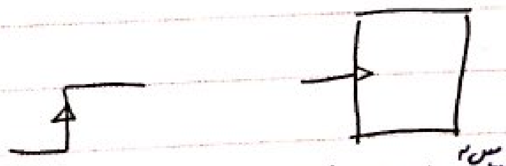
D_2: روشن Q_2 = آخرین D_1



حساس به سطح منفی



حساس به سطح مثبت



حساس به لبه بالا رونده

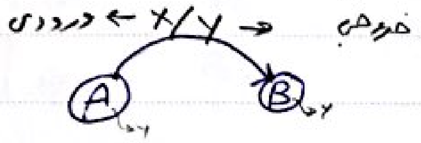
حساس به لبه پایین رونده

Subject
Date

تحلیل مدار :

تحلیل مدار مستلزم معرفی یک state diag است . state Diag شامل یک سری دایره است .

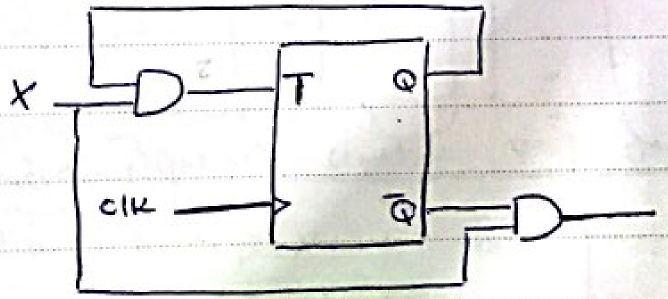
یعنی به تعداد state ها دایره داریم . و با کمان وضعیت ها را بهم وصل می‌کنیم .



خروجی را روی کمان ← میلی

خروجی را داخل دایره ها نشان می‌دهیم ← مور

EX :



① تشخیص ورودی و خروجی :

ورودی ← $X, Q(t)$

خروجی ← $Z, Q(t+1)$

② $Z = X.\bar{Q}(t)$

$Q(t+1) = T \oplus Q(t)$ ← معادله خروجی T_ff

$T = XQ(t)$

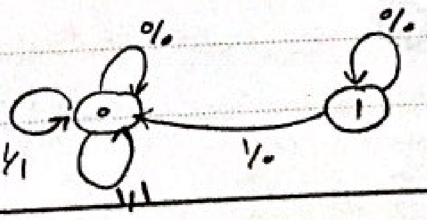
$Q(t+1) = XQ(t) \oplus Q(t)$

③ جدول

ورودی		خروجی	
X	Q(t)	Z	Q(t+1)
0	0	0	0
0	1	0	1
1	0	1	0
1	1	0	0

④ جایگذاری ورودی ها

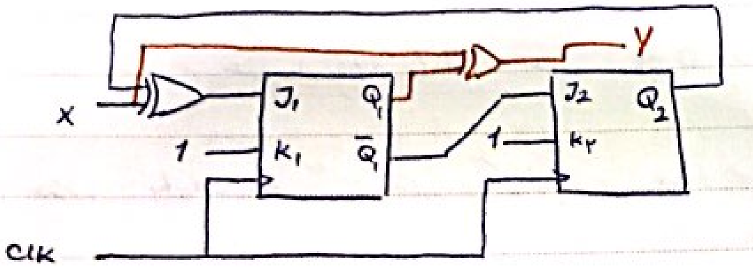
⑤ رسم دیاگرام



Subject

Date

Ex: برای مدار ترتیبی زیر state Diag رسم کنید.



$$Q(t+1) = J\bar{Q}(t) + \bar{K}Q(t)$$

ورودی : X , $Q_1(t)$, $Q_2(t)$

خروجی : Y , $Q_1(t+1)$ و $Q_2(t+1)$

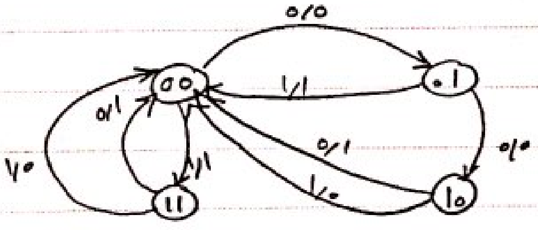
$$Y = X \oplus Q_1(t)$$

$$Q_1(t+1) = (X \oplus Q_2(t)) \cdot \bar{Q}(t) + \cancel{0 \cdot Q_1(t)}$$

$$Q_2(t+1) = \bar{Q}_1(t)\,\bar{Q}_2(t) + \cancel{0 \cdot Q_2(t)}$$

X	$Q_1(t)$	$Q_2(t)$	Y	$Q_1(t+1)$	$Q_2(t+1)$
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	1	0	0
1	0	0	1	1	1
1	0	1	1	0	0
1	1	0	0	0	0
1	1	1	0	0	0

(شل حعم ۰ فرق داره ۱.)

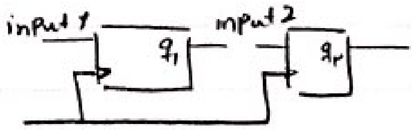


Subject

Date

حالت برعکس و از روی state Diag مدار ترتیبی تنظیم را رسم کنید.

state	input 0	input 1
A	A/0	B/0
B	C/0	B/0
C	D/0	B/0
D	B/1	A/0



log تعداد حالت های تعداد ff

state های کاراکتری بود باید شماره بزنیم

A=00 B=01 C=10 D=11

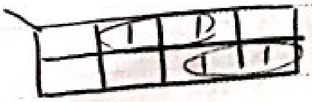
② → جدول →

(شماره state; جدول تحریک)

X	$Q_1(t)$	$Q_2(t)$	Y	$Q_1(t+1)$	$Q_2(t+1)$	T_1	D_2
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	0	1	1	0	1
0	1	1	1	0	1	1	1
1	0	0	0	0	1	0	1
1	0	1	0	0	1	0	1
1	1	0	0	0	1	1	1
1	1	1	0	0	0	1	0

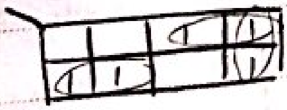
$Y = \Sigma m(3) = \bar{x}\, Q_1(t)\, Q_2(t)$

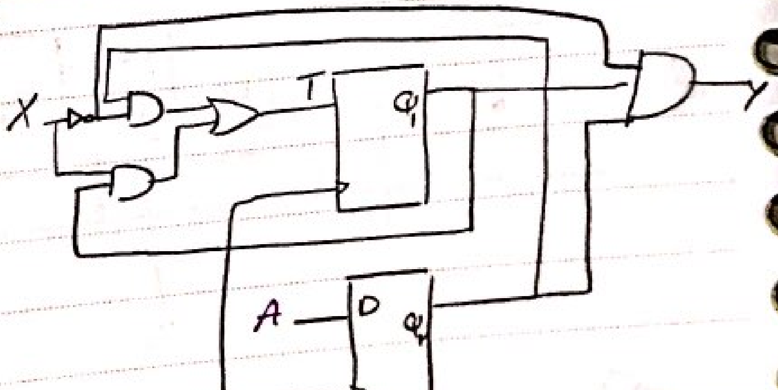
$T_1 = \Sigma m(1,3,6,7) = \bar{x} Q_2(t) + x Q_1(t)$



$D_2 = \Sigma m(2,3,4,5,6) = \bar{x} Q_1(t) + Q_1(t)\bar{Q}_2(t) + x\bar{Q}_1(t)$

A

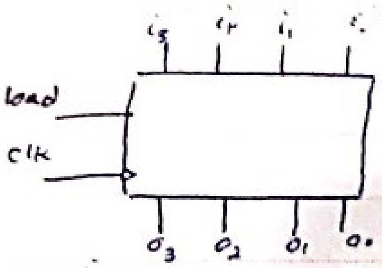




Scanned by CamScanner

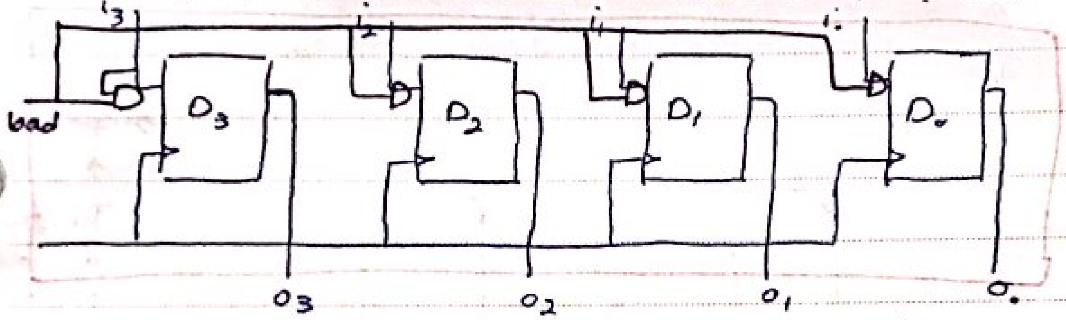
Subject
Date

Register ← جایی هستند که داده ها در آن ذخیره هستند



داخل رجیستر از D-ff استفاده شده است چون دومیم

ار ورودی هست در خروجی هم هست.



وقتی چندتا رجیستر داشته باشیم و بخواهیم معلوم کنیم توی کدوم خلیه مقادیر نوشته شه از load استفاده می کنیم.

shift_Register

10 11 0011

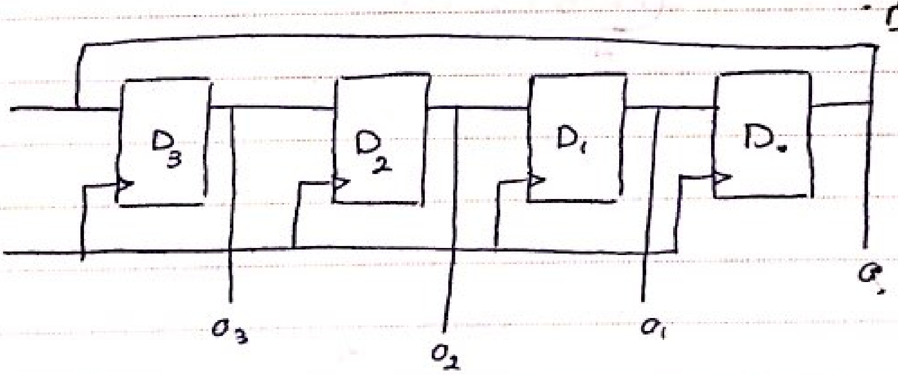
shR → 01011001

shR 11011001

shL 01100111

EX رجیستری طراحی کنید که قابلیت شیفت دادن داشته باشد. (4 bit shift Register)

نیاز به ۴ D-ff داریم.



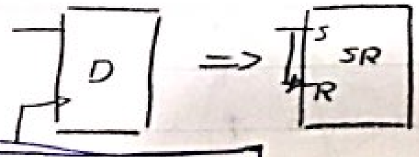
Subject
Date

شمارنده ها (Counter) ← جانسون

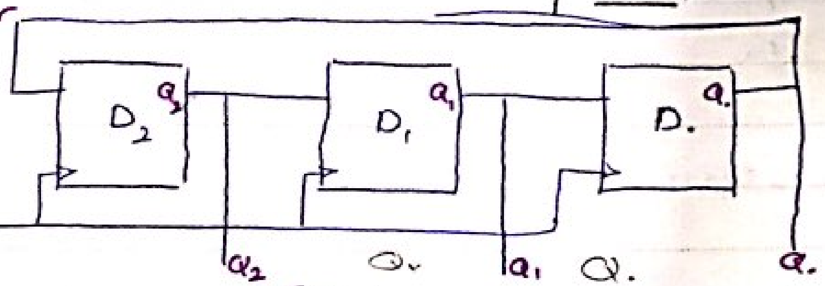
شمارنده حلقوی (ring counter)

ستسی بر مبنای یک شمارنده ← ها

چند تایی باشه چند فلیپ فلاپ لازم داریم



Ring counter



مانند شمارش ۳؟
شمارنده ۳ تایی

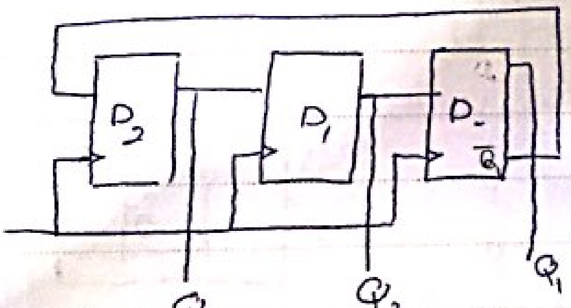
① تشخیص ورودی خروجی
② معادله

		Q_2	Q_1	Q_0
$Q_0(t+1)=Q_1(t)$	حالت اولیه	1	0	0
$Q_1(t+1)=Q_2(t)$		0	1	0
$Q_2(t+1)=Q_0(t)$		0	0	1

$Q(t+1)=0$

جانسون

بعد از چند پالس دوباره به حالت اولیه می‌رسیم؟



پس از ۶ پالس

	Q_3	Q_2	Q_1
حالت اولیه	0	0	0
$Q_1(t+1)=Q_2(t)$	1	0	0
$Q_2(t+1)=Q_3(t)$	1	1	0
$Q_3(t+1)=\overline{Q}_1(t)$	1	1	1
	0	1	1
	0	0	1
	0	0	0

Scanned by CamScanner

Subject
Date

۲ log تعداد state ها فلیپ فلاپ لازم داریم

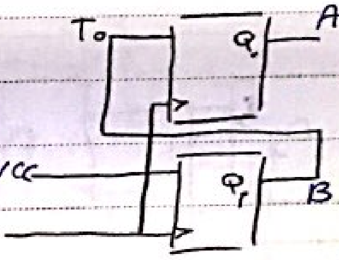
طراحی

(شکل حل ه فرق داره)

۱) رسم جدول حالت قبلی و فعلی شمارنده ۲ بیتی صعودی شمار؟

حالت فعلی q_0	q_1	حالت بعدی q_0^+	q_1^+	T_0	T_1
0	0	0	1	0	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	0	0	1	1

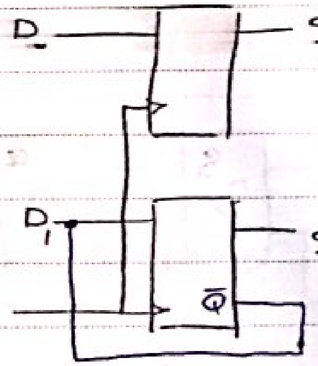
$T_0 = \bar{q}_0 q_1 + q_0 q_1 = q_1(t)$



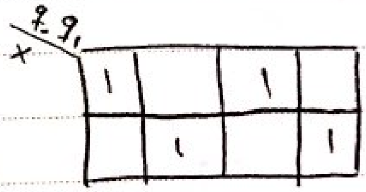
EX: شمارنده با پایه کنترلی طراحی کنید. که پایه کنترلی ۰ باشه نزولی شمارد و ۱ باشد صعودی

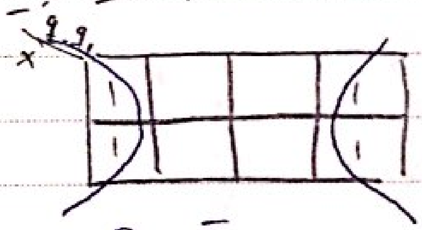
جدول تحریک D هر چی الان باشه همون میشه.

x	q_0	q_1	q_0^+ (D_0)	q_1^+ (D_1)	D_0	D_1
0	0	0	1	1	1	1
0	0	1	0	0	0	0
0	1	0	0	1	0	1
0	1	1	1	0	1	0
1	0	0	0	1	0	1
1	0	1	1	0	1	0
1	1	0	1	1	1	1
1	1	1	0	0	0	0



حالا براساس جدول تحریک نزولی D_0 و





$D_0 = \bar{x}\bar{q}_0\bar{q}_1 + \bar{x} q_0 q_1 + x \bar{q}_0 q_1 + x q_0 \bar{q}_1$

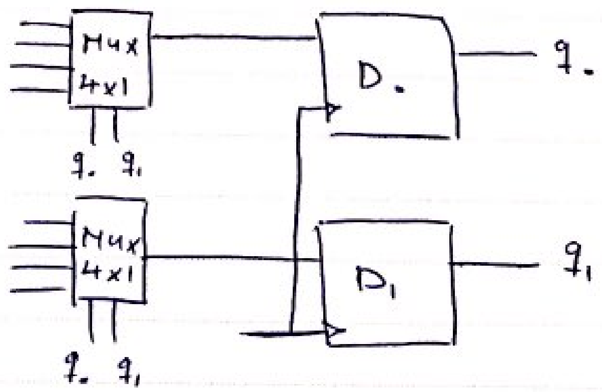
$D_1 = \bar{q}_1$

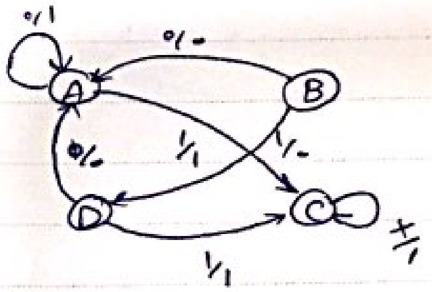
* طراحی از روی معادله D_0 و D_1

Scanned by CamScanner

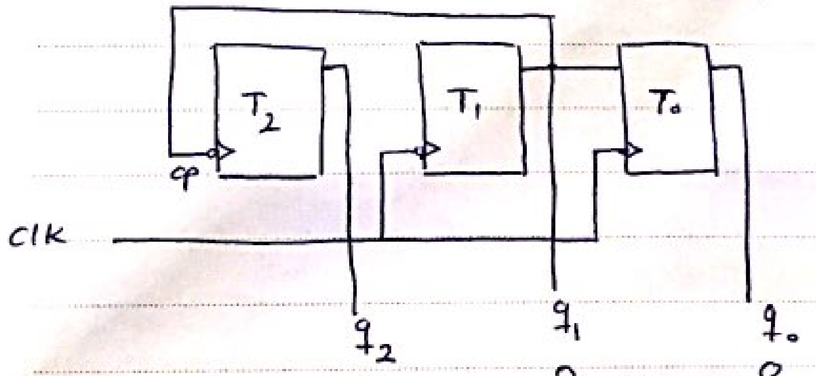
Subject
Date

Ex: ورودی Mux ۴×۲ او ۲ را به گونه مشخص کنید که مدار ترتیبی زیر نمودار حالت روبرو را داشته باشد.





پالس ها با هم برابر نیست ← آسنکرون



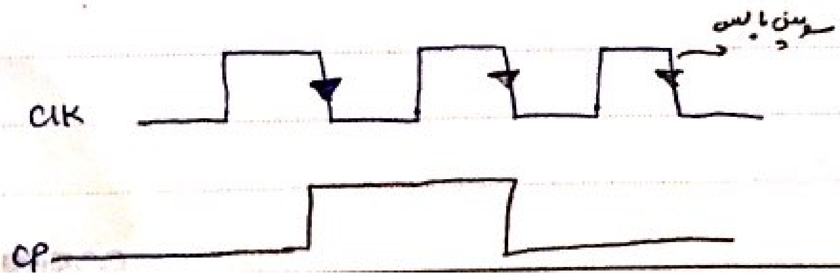
q_2	q_1	q_0
0	0	0
0	1	0
0	0	1
1	0	1
1	1	1

T_2	T_1	T_0
off	on	on
off	on	on
on	of	off
off	on	n

$Q_0(t+1)=Q_1(t)\oplus Q_0(t)$

$Q_1(t+1)=1\oplus Q_1(t)=\overline{Q}_1(t)$

$Q_2(t+1)=1\oplus Q_2(t)=\overline{Q}_2(t)$



Scanned by CamScanner